Mechatronics 21 (2011) 560–569

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Design optimization on the drive train of a light-weight robotic arm

Lelai Zhou [a,*], Shaoping Bai [a], Michael Rygaard Hansen [b]

[a] Department of Mechanical and Manufacturing Engineering, Aalborg University, Pontoppidanstræde 103, 9220 Aalborg, Denmark
[b] Department of Engineering, University of Agder, Grooseveien 36, 4876 Grimstad, Norway

ARTICLE INFO

Article history:
Received 15 April 2010
Accepted 7 February 2011
Available online 2 March 2011

Keywords:
Drive train optimization
Discrete design variables
Light-weight robot
Complex method

ABSTRACT

A drive train optimization method for design of light-weight robots is proposed. Optimal selections of motors and gearboxes from a limited catalog of commercially available components are done simultaneously for all joints of a robotic arm. Characteristics of the motor and gearbox, including gear ratio, gear inertia, motor inertia, and gear efficiency, are considered in the drive train modeling. A co-simulation method is developed for dynamic simulation of the arm. A design example is included to demonstrate the proposed design optimization method.

1. Introduction

The drive train is the core part of a robot system, with significant impact on the cost and performance of the whole system. To achieve a light-weight design, drive train optimization plays a key role. A number of methods for motor and gear selection in mechatronic systems have been proposed. Pasch and Seering [1] studied maximizing the system acceleration by optimal selection of transmission ratio. van De Straete et al. [2,3] proposed a general method of motor and gearbox selection for optimization of servo drive system. The method automates the solution procedure for the servo drive design problem by virtue of the normalization of torques, velocities, and transmission ratios. Cetinkunt [4] proposed an optimization approach of balancing the high speed and precision in servo systems. Cusimano [5,6] presented a procedure for optimal selection of an electrical motor and transmission. Roos et al. [7] proposed a method of finding the best motor/gear ratio combination for any given load with respect to weight, size, peak power, torque and efficiency. The methods above are applicable to the design of a single joint combining a motor and a gearbox, and they do not address the discrete nature of the selection process.

For the design of robotic drive train consisting of multiple joints, the challenge is that not only the characteristics of motor and gearbox at a single joint, but also the dynamics of the robot should be taken into account, the latter varying with the selection of components and link dimensions. Furthermore, the optimization procedure adopted has to be capable of handling discrete design variables because the transmission is typically composed of commercially available components. Very few methods are available for the optimization of the entire drive train of a robot under constraint of available components. A method for the optimum selection of robot actuators was proposed in [8], with objective to minimize the total mass of all the actuators under torque and temperature constraints. Pettersson and Ölvander [9] reported recently a method of design optimization, in which drive train for two joints were optimized for an industrial manipulator. The method is not applicable to selection of components from a catalog. An evolutionary approach of optimization on robot configurations was reported in [10]. A simulation environment called Modelica with robot optimization characteristic was presented in [11], where the parameters of a controller can be tuned by a multi-criteria parameter optimization method to improve the system dynamics. DLR's 7-dof (degrees of freedom) torque-controlled light-weight robotic arm was built with customized motors and gearboxes to achieve a low weight [12]. Methods of robot optimization can also be found in [13–15], among others.

In this paper, an optimization method for drive train design of a light-weight robotic arm is proposed. The method is applicable to serial robotic arms, aiming at minimizing the arm weight. In the method, the optimization is carried out with a prescribed trajectory of the end-effector, generated within the robotic arm's workspace. Moreover, the inverse kinematic analysis was conducted in ADAMS to verify that the trajectory is within the joint space. A dynamic model of the robotic arm is developed, upon which an optimization problem is formulated. A non-gradient optimization method, namely, the Complex [16], is implemented to run the optimization. The method is implemented on a co-simulation platform, where robotic dynamics is determined using MSC.ADAMS™, and the complex optimization is performed in Matlab™.

* Corresponding author. Tel.: +45 9940 3350; fax: +45 9815 1675.
E-mail address: lzh@m-tech.aau.dk (Lelai Zhou).

0957-4158/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechatronics.2011.02.004

Nomenclature

ξ	vector of gravitational forces
$\ddot{\theta}(t)$	joint acceleration (rad/s^2)
$\dot{\theta}(t)$	joint speed (rad/s)
η_g	gear efficiency
$\mathbf{M}$	mass matrix
$\mathbf{q}$	position vector
$\mathbf{R}$	rotation matrix
$\mathbf{u}_m(\mathbf{u}_g)$	index numbers for motors (gearboxes)
$\mathbf{v}$	vector of Coriolis and centrifugal terms
$\mathbf{x}_b(\mathbf{x}_w)$	best (worst) point
$\mathbf{x}_c(\mathbf{x}_{cand})$	centroid (candidate) point
ρ	gear ratio
$\tau(t)$	required joint torque (Nm)
$J_m(J_g)$	motor (gear) inertia (kg m^2)
$m_m(m_g)$	motor (gear) mass (kg)
N_g^{max}	max permissible input speed of gearbox (rpm)
N_m^{max}	max permissible motor speed (rpm)
n_p	required motor peak speed (rpm)
n_{in}	required max input peak speed (rpm)
T_g^{max}	limit for momentary peak torque of gearbox (Nm)
T_m^{max}	motor stall torque (Nm)
T_g	limit for rated torque of gearbox (Nm)
T_m	motor nominal torque (Nm)
τ_g	required gear peak torque (Nm)
τ_m	required motor torque (Nm)
τ_p	required motor peak torque (Nm)
τ_{rmc}	RMC value of required joint torque (Nm)
τ_{rms}	RMS value of required motor torque (Nm)

2. Conceptual design of a robotic arm

The light-weight robotic arm considered in this paper has five degrees of freedom (dof), with two dof at the shoulder, one at the elbow, and two at the wrist, as depicted in Fig. 1. The arm is designed for assisting elderly and handicapped people in daily living [17]. Light-weight design of such robotic arms is required for safety and energy efficiency.

In this design, harmonic drives are used as gearing elements. The motors and harmonic drive gearboxes are mounted inside the joints, while the axes of rotation coincide with the joint axes. The physical realization of Joint 2 is illustrated in Fig. 2. The same conceptual design is used for all 5 joints.

While the topology of the individual transmission is fixed, the motors can be chosen from either permanent magnet DC motors or brushless DC motors. Maxon™ motors are used in this study. The gearboxes are limited to Harmonic Drive™ backlash-free coaxial gears. Both components are considered appropriate for implementing the proposed design optimization method, that may easily accommodate a wider variety of gearboxes and motors. The arm structures are made of aluminium.

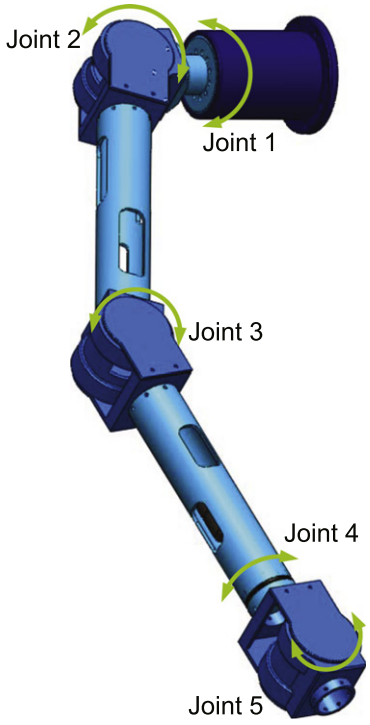


Fig. 1. A 5-dof light-weight robotic arm.

3. Kinematics and dynamics

3.1. Kinematics

The forward kinematics of the robotic arm is formulated based on the Denavit–Hartenberg (D-H) convention [18]. A Cartesian coordinate system is attached to each link of the robotic arm, as shown in Fig. 3. D–H parameters are defined as listed in Table 1. The detailed solution of the forward kinematics can be found in Appendix A.

For given locations of the end-effector, the joint variables are found by inverse kinematics. The method presented in [19] was adopted for this purpose. The detailed solution of the inverse kinematics can be found in Appendix B.

3.2. Inverse dynamics

The computation of the inverse dynamics is a prerequisite for evaluating any given design with given load and prescribed trajectory. Here we briefly recall the Lagrange–Euler formulation, which is

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} = \tau_i; \quad i = 1, \ldots, 5. \tag{1}$$

where the Lagrangian $L = K - U = \sum_{i=1}^{5}(K_i - U_i)$. For the ith link, the kinetic energy K_i and the potential energy U_i are given by

$$K_i = \frac{1}{2} m_i \mathbf{v}_{ci}^T \mathbf{v}_{ci} + \frac{1}{2} \boldsymbol{\omega}_i^T \mathbf{I}_i \boldsymbol{\omega}_i; \quad U_i = m_i \mathbf{g}^T \mathbf{p}_{ci} \tag{2}$$

where $\mathbf{v}_{ci}$ denotes the linear velocity of the center of mass for link i, $\boldsymbol{\omega}_i$ is the angular velocity of the same link, and $\mathbf{I}_i$ is the inertia matrix of link i with respect to its center of mass. Moreover, $\mathbf{p}_{ci}$ is the position vector of the center of mass for link i, measured in the reference coordinate system.

Substituting Eq. (2) into (1) produces equations of motion as

$$\mathbf{M}(\theta_i)\ddot{\theta}_i + \mathbf{v}(\theta_i, \dot{\theta}_i) + \boldsymbol{\xi}(\theta_i) = \boldsymbol{\tau} \tag{3}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{v}$ is the vector of Coriolis and centrifugal terms of the links, $\boldsymbol{\xi}$ is the vector of gravitational forces, and $\boldsymbol{\tau}$ is the vector of joint torques.

Eq. (3) can be solved with different approaches [20,21]. In this work, the dynamics solutions are found through ADAMS, which

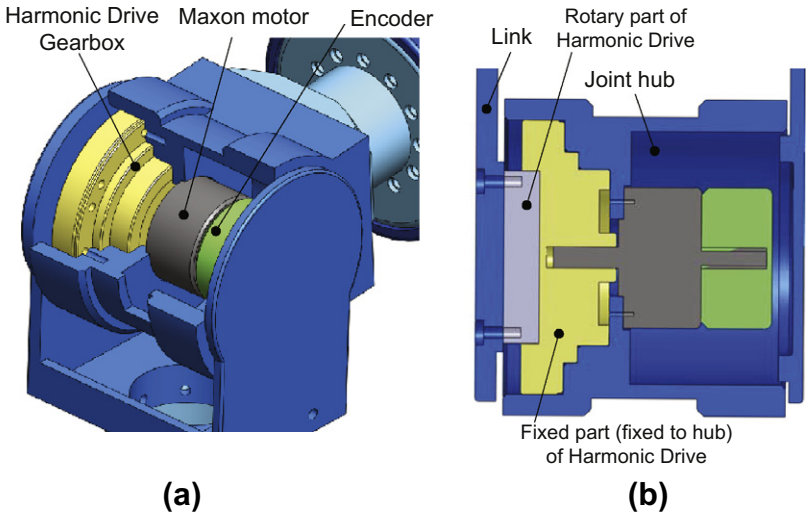


Fig. 2. Joint mechanism for Joint 2, (a) 3D view, (b) section view.

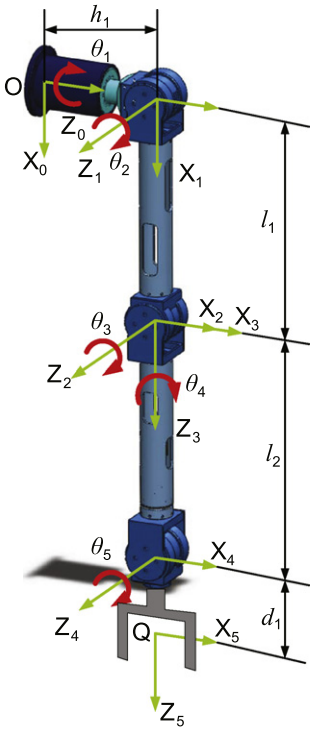


Fig. 3. Robotic arm coordinate system.

Table 1
D–H Parameters of the robotic arm.

Joint i	α_i	a_i	d_i	θ_i
1	$\pi/2$	0	h_1	θ_1
2	0	l_1	0	θ_2
3	$\pi/2$	0	0	θ_3
4	$-\pi/2$	0	l_2	θ_4
5	$\pi/2$	0	d_1	θ_5

directly takes advantage of the accurate geometry and mass property of a CAD embodiment for computations. In the meantime, a Matlab solver adopted the recursive approach [22] was also developed for the purpose of comparison, which is discussed in Section 6.

3.3. Drive train modeling

Eq. (3) yields the required joint torque $\tau(t)$, if the motion is prescribed. The motor torque for each joint can further be determined, as seen in Fig. 4. For the harmonic drive gearbox, the gear efficiency varies depending on the output torque. With the inertia of motor and gear, the required motor torque for the ith joint is derived as

$$\tau_{m,i} = \left\{ (J_m + J_g)\ddot{\theta}(t)\rho + \frac{\tau(t)}{\rho\eta_g} \right\}_i ; \quad i = 1, \ldots, 5 \tag{4}$$

where ρ_i is the gear ratio, $J_{g,i}$ is the gear inertia with respect to the input motor axis, $J_{m,i}$ is the motor inertia, and $\eta_{g,i}$ is the gear efficiency.

4. Formulation of design problems

The criteria for selecting motor and gearbox are applicable to each single joint, thus subscript 'i' is omitted in this section for clarity.

4.1. Motor selection criteria

Motors for robotic arms are usually selected from two motor groups, brushed and brushless DC motors. In selecting motors, the following three constraints must not be violated:

Nominal torque limit. The nominal torque is the so-called maximum continuous torque. The root mean square (RMS) value τ_{rms} of the required motor torque τ_m has to be smaller than or equal to the nominal torque of the motor T_m

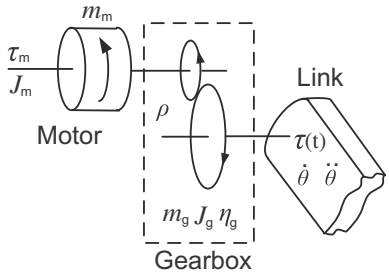


Fig. 4. Schematic view of drive train model for a single joint.

$$\tau_{rms} \leqslant T_m \tag{5}$$

where $\tau_{rms} = \sqrt{\frac{1}{\Delta t}\int_0^{\Delta t} \tau_m^2 dt}$, with Δt being the duration of a characteristic working cycle.

Stall torque limit. The stall torque is the peak torque of the motor. The required peak torque τ_p has to be smaller than or equal to the stall torque T_m^{max} of the motor

$$\tau_p \leqslant T_m^{max} \tag{6}$$

where $\tau_p = \max\{|\tau_m|\}$.

Maximum permissible speed limit. The maximum permissible speed for DC motors is primarily limited by the commutation system. A further reason for limiting the speed is the rotor's residual mechanical imbalance which shortens the service life of the bearings. The required peak speed n_p corresponding to the motor has to be smaller than or equal to the maximum permissible speed N_m^{max} of the motor

$$n_p \leqslant N_m^{max} \tag{7}$$

where $n_p = \max\{|2\pi\dot{\theta}(t)\cdot\rho|\}$.

The inequalities (5)–(7) represent the constraints that must be fulfilled by any motor in the drive train.

4.2. Gearbox selection criteria

In the selection of gearboxes, the following three constraints are considered:

Rated output torque limit. It is recommended by the Harmonic Drive gearbox manufacturer to use the RMC value for calculating rated torque [23]. The RMC value is a measure of the accumulated fatigue on a structural component and reflects typical endurance curves of steel and aluminium [24]. It is therefore relevant to gearbox lifetime, and this criterion has also been used in robotic applications [25]. With this criterion, a constraint is derived as

$$\tau_{rmc} \leqslant T_g \tag{8}$$

where $\tau_{rmc} = \sqrt[3]{\frac{1}{\Delta t}\int_0^{\Delta t} \tau^3(t) dt}$, with $\tau(t)$ being the required torque from the gearbox output. T_g is the limit for rated torque of the gearbox.

Maximum output torque limit. The required peak torque τ_g with respect to the output side has to be smaller than or equal to the allowable peak torque T_g^{max} of the harmonic drive

$$\tau_g \leqslant T_g^{max} \tag{9}$$

where $\tau_g = \max\{|\tau(t)|\}$.

Maximum permissible input speed limit. The required maximum input peak speed n_{in} has to be smaller than or equal to the maximum permissible input speed N_g^{max} of a gearbox

$$n_{in} \leqslant N_g^{max} \tag{10}$$

where $n_{in} = \max\{|\dot{\theta}(t)\cdot\rho|\}$.

The inequalities (8)–(10) represent the constraints that must be fulfilled by any gearbox in the drive train.

Although the inequalities (5)–(10) are derived specifically from the selection criteria for the motors and gearboxes considered in this paper, they are quite general, and would be recognized in any selection procedure for motors and gearboxes suitable for robotic arm design.

4.3. Objective function formulation

The objective of the optimization is to minimize the mass of the robotic arm. In this formulation, we minimize only the mass of the power transmission, while the mass of the arm structures (m_{arm}) remains constant. Therefore, the optimization task is to find the lightest combination of motor and gearbox for all five dof that fulfill all constraints associated with the motors and gearboxes. The objective function, $f(\mathbf{x})$, is defined as the sum of the mass of the motors and gears, as shown in Eq. (11a).

$$\min_{\mathbf{x}} f(\mathbf{x}) = \sum_{i=1}^{5}\{m_m(\mathbf{u}_m) + m_g(\mathbf{u}_g)\}_i \tag{11a}$$

$$\mathbf{x} = [\mathbf{u}_m, \mathbf{u}_g]$$

S.T.

$$T_{m,i} \geqslant \sqrt{\frac{1}{\Delta t}\int_0^{\Delta t}\left\{(J_m(\mathbf{x}) + J_g(\mathbf{x}))\ddot{\theta}(t)\rho + \frac{\tau(t,\mathbf{x})}{\rho\eta_g}\right\}_i^2 \cdot dt} \tag{11b}$$

$$T_{m,i}^{max} \geqslant \max\left\{\left|(J_m(\mathbf{x}) + J_g(\mathbf{x}))\ddot{\theta}(t)\rho + \frac{\tau(t,\mathbf{x})}{\rho\eta_g}\right|\right\}_i \tag{11c}$$

$$N_{m,i}^{max} \geqslant \max\left\{|2\pi\dot{\theta}(t)\cdot\rho|\right\}_i \tag{11d}$$

$$T_{g,i} \geqslant \sqrt[3]{\frac{1}{\Delta t}\int_0^{\Delta t}\tau_i^3(t,\mathbf{x})\cdot dt} \tag{11e}$$

$$T_{g,i}^{max} \geqslant \max\{|\tau(t,\mathbf{x})|\}_i \tag{11f}$$

$$N_{g,i}^{max} \geqslant \max\left\{|\dot{\theta}(t)\cdot\rho|\right\}_i \tag{11g}$$

where design variables $\mathbf{x}$ includes the index numbers of motors $\mathbf{u}_m = [u_{m1},\ldots,u_{m5}]$ and gearboxes $\mathbf{u}_g = [u_{g1},\ldots,u_{g5}]$, relative to databases containing commercially available components. So far, we have formulated the design problem as a discrete optimization problem, which can be solved by commercially available codes. We select a non-gradient method called Complex for this purpose. The implementation is outlined in the next section.

5. Procedure of optimization

The optimization method is developed as a Matlab and MSC.ADAMS co-simulation platform. The optimization algorithm is based on the Complex method, which is briefly discussed.

5.1. Optimization by complex

The Complex method is a non-gradient based optimization method, first presented by Box [16].

In the Complex method, several possible designs (design population) are manipulated. The method is based on a feasible domain, containing a design population as a set of design points. The number of design points has to be greater than the number of independent design variables. The starting design points (initial population) are randomly generated, and evaluated through the objective function to check performance and constraint violation. Among all populations, the set of design variables having the minimal objective function is denoted as the best point $\mathbf{x}_b$, while the one having the maximal objective function is denoted as the worst point $\mathbf{x}_w$. Their corresponding values of objective function are noted as the best and worst values. The centroid point is calculated as

$$\mathbf{x}_c = \frac{1}{m-1}\sum_{i=1}^{m}\mathbf{x}_i, \quad \mathbf{x}_i \neq \mathbf{x}_b \tag{12}$$

$$\mathbf{x}_i = [x_1, x_2, \ldots, x_n], \quad m > n \tag{13}$$

The main idea of the Complex method is to replace the worst point by a new and better point. The new point is found by the reflection of the worst point through the centroid with a reflection coefficient α, yielding the following expression for the new design point

$$\mathbf{x}_{cand} = \mathbf{x}_c + \alpha(\mathbf{x}_c - \mathbf{x}_w) \tag{14}$$

The coefficient $\alpha = 1.3$ is used in this study, as recommended in [16]. The candidate point $\mathbf{x}_{cand}$ is checked through explicit and implicit constraints. When it conforms to the constraints, $\mathbf{x}_{cand}$ replaces $\mathbf{x}_w$. This method cannot handle the situation when the centroid is trapped in a local minimum. Therefore, the method has been modified such that the point moves towards the best point if it continues to be the worst one. To avoid the collapse of the algorithm, a random value is also added to the new point. The modified method to calculate the reflection point is given as

$$\mathbf{x}_{cand}^{new} = \frac{1}{2}\left(\mathbf{x}_{cand}^{old} + \varepsilon\mathbf{x}_c + (1-\varepsilon)\mathbf{x}_b\right) + (\mathbf{x}_c - \mathbf{x}_b)(1-\varepsilon)(2k-1) \quad (15)$$

where k is a random number varying in the interval [0,1], with

$$\varepsilon = \left(\frac{n_r}{n_r + k_r - 1}\right)^{\frac{n_r + k_r - 1}{n_r}} \quad (16)$$

Here k_r is the number of times the same point has repeatedly been identified as the worst point, and n_r is a tuning parameter which is set to 4. The convergence criterion of the Complex method in this work is that the difference between the best and worst objective function values is less than a user defined tolerance.

5.2. Dynamics model with MSC.ADAMS

The drive requirements of the whole robotic arm system are determined from inverse kinematic and dynamic analysis within MSC.ADAMS. The inverse kinematic and dynamic analysis is developed as a simulation package, which will be called by the optimization program. To this end, the mass of motors and gearboxes are parameterized, while the trajectory of the robotic arm is prescribed. For each variation of motors and gearboxes, the required motor torques are accurately calculated. The mass of distribution is updated during the optimization procedure.

The inverse kinematic and dynamic analysis of the robotic arm in ADAMS follows a so-called master-slave approach, as shown in Fig. 5. The basic concept of this approach is that we make two models of the robotic arm in ADAMS, a master model and a slave one. In the master model, the inverse kinematic analysis is executed to record the joint motions corresponding to the prescribed end-effector trajectory. In the slave model, the joint motion data is imported and imposed on the joints, and payload is also attached to the end-effector. Then the inverse dynamic calculation is performed to solve the required joint torques for actuating the robotic arm.

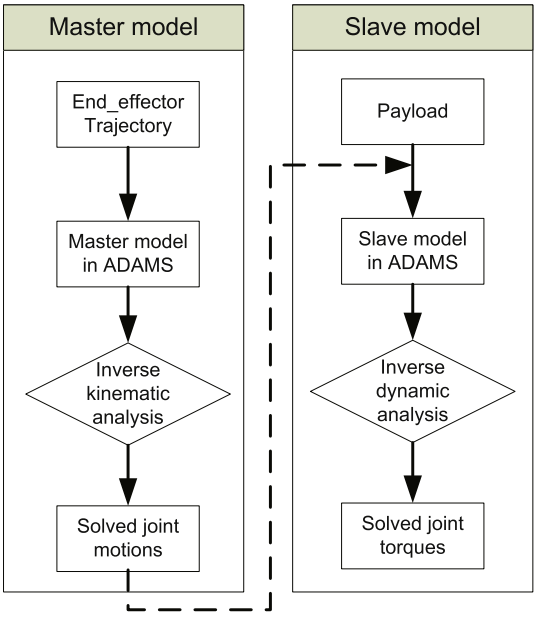


Fig. 5. The procedure of inverse kinematic and dynamic analysis.

In the master-slave approach, we can define different trajectories and payloads for the robotic arm model, which makes the model more flexible for different simulation conditions. This approach can be applicable to other serial and parallel robot systems.

5.3. Matlab-ADAMS co-simulation platform

The design optimization is mainly concerned of two tasks: the optimization routine and creation of a parametric dynamic simulation model. Both tasks can be performed on a Matlab-ADAMS co-simulation platform developed in this work. As shown in Fig. 6, the platform works with two modules. The ADAMS module is used to simulate the inverse kinematics and dynamics of the robotic arm. The Matlab module implements the Complex method to call the ADAMS simulation in batch mode.

6. An example of design optimization

Design optimization was conducted on the 5-dof light-weight robotic arm. The link lengths of the robotic arm are fixed. The trajectory of the end-effector in the base coordinate system is defined as $X_{ef}(t) = 50 + 400(1 - \cos(t))$, $Y_{ef}(t) = -1000 + 800(1 - \cos(t/2))$, and $Z_{ef}(t) = 280 + 250(\cos(t/2) - 1)$, all with unit of *mm*. The corresponding velocity and acceleration profiles of the trajectory are depicted in Fig. 7. The Euler angles for the end-effector are given as $[\sin(t/180), 0, 0]$, which implies the end-effector remains horizontal during the prescribed motion. The motion of the end-effector is illustrated in Fig. 8.

The payload is defined as a mass point in ADAMS and weights 2 kg. On the other hand, the mass of motors and gearbox are determined from their indices selected. The solved motions of each joint, as shown in Fig. 9 for Joint 1, are imported into ADAMS to generate arm dynamics.

Ten candidate motors from the Maxon Motor catalog are arranged ascendingly with respect to the mass of motor, as shown in Table 2.

The gearboxes used in the robotic arm are selected from harmonic drive CPU units, as listed in Table 3. For the harmonic drive gearboxes, the efficiency is a function of operation speed. In this paper, the gear efficiency is set as $\eta_g = 0.85$ for all gearboxes, for simplicity.

In order to simplify the process of selecting motors and gearboxes, the gear ratio of each joint is fixed as $\boldsymbol{\rho} = [344, 444, 100, 51, 100]$, orderly from Joints 1–5. The gear ratio is based on previous investigation of joint torques. The ratios ρ_1 and ρ_2 are combinations of two gearboxes, a planetary gearhead and a harmonic drive unit, to achieve high gear ratio. For simplicity, the mass of the planetary gearhead is fixed, while only the mass of the harmonic drive gearbox is manipulated.

The gearbox is selected for each joint, associated with the selection of motor. The harmonic drive CPU unit is used in all joints except Joint 4, due to the joint structure consideration. A planetary gearhead is used in Joint 4, so $u_{g4} = 0$.

6.1. Optimization results

An optimized design of motor and gearbox for the robotic arm was found, as listed in Table 4. The optimized weight of the robotic arm is 10.2 kg, with a reduction of 38% corresponding to the initial combination of motors and gearboxes.

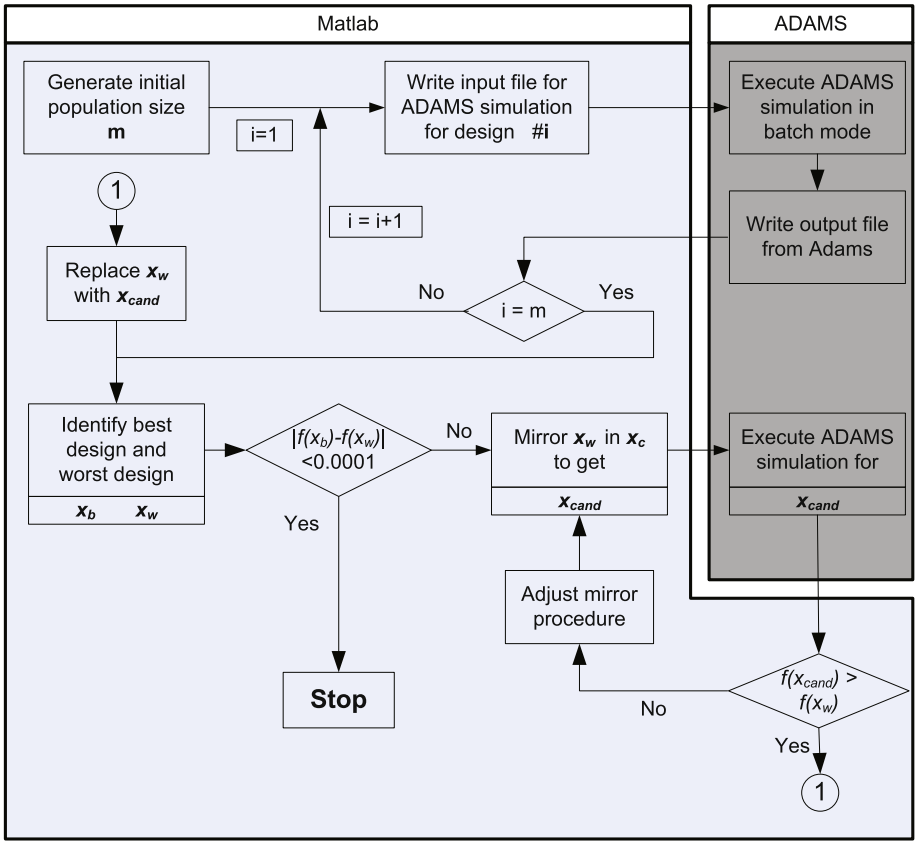


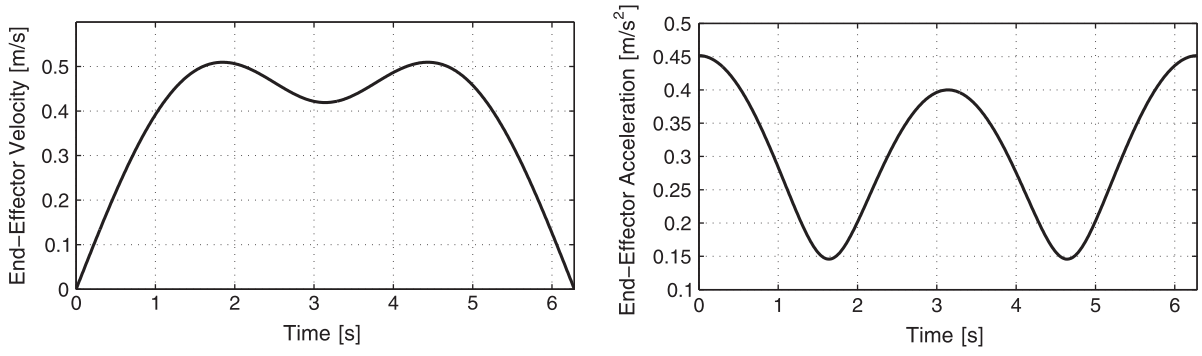
Fig. 6. Diagram of the optimization routine in the co-simulation platform.

The convergence of the objective function is depicted in Fig. 10, where both the best and the worst objective function values from the Complex algorithm are displayed. The solution to the optimal result is achieved after 3160 iterations with a population size of 140. In this work, the tolerance of convergence criterion is set to 0.0001. It is noted from Fig. 10 that at the 1500th iteration, the difference between the best and the worst $f(\mathbf{x})$ values is 0.03, which means the convergence criterion is not met at that point, even though two values appear to be very closer.

Fig. 11a illustrates the convergence of the motor design variables. Note that only convergence curves for Joints 1 and 5 are displayed for clarity. The convergence of the gearbox design variables is depicted in Fig. 11b. Comparing the convergence rate for the motor and gearbox design variables, the gearbox design variables converge faster than the motor design variables. A possible explanation is the mass difference between the harmonic drive units is larger than that between the motors.

Based on the optimization results, the motor torques are obtained for Joints 1, 2, 3 and 5, as shown in Fig. 12, where torques for initial designs are also displayed for comparison. It is seen that the optimal design reduces the peak torque by 31.8% reduction for Joint 1 and by 40% for Joint 2.

To verify the accuracy of the solved joint torques from the co-simulation platform, another program was developed for simulation with Matlab only. The joint torques obtained with the two methods are shown in Fig. 13 for the same robot trajectory. Higher torques calculated by the co-simulation platform were observed for both Joints 1 and 2. The reason is that ADAMS in the co-



(a) Velocity profile (b) Acceleration profile

Fig. 7. Velocity and acceleration of the end-effector trajectory.

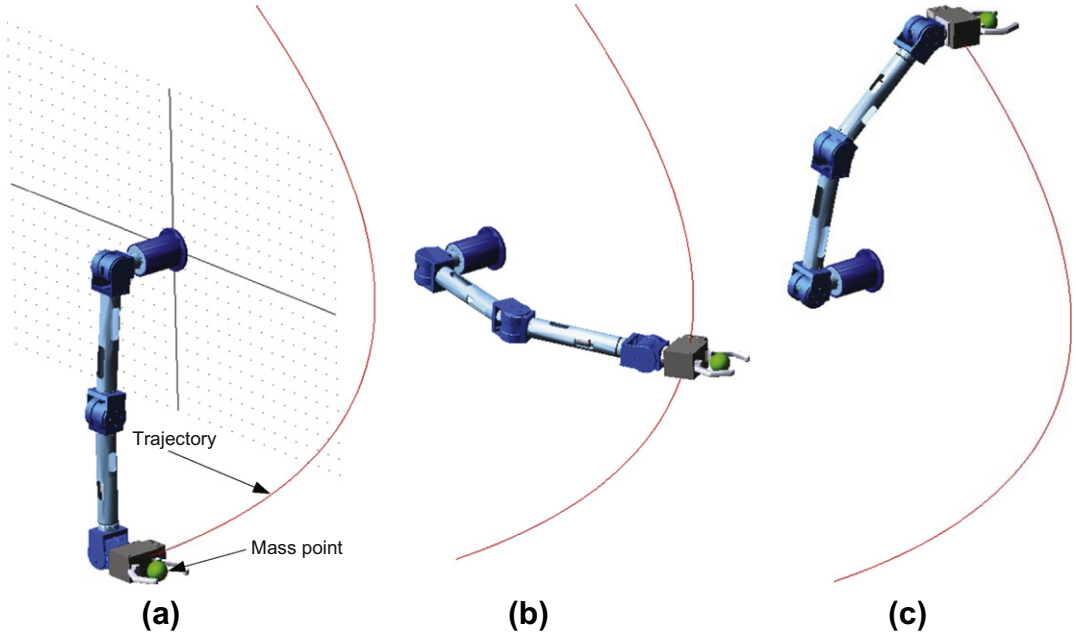
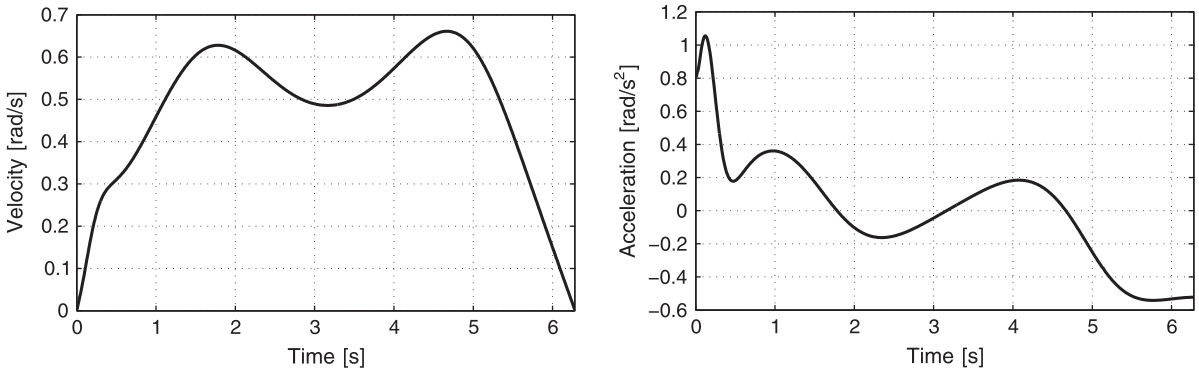


Fig. 8. Illustration of a prescribed end-effector motion.



(a) Velocity profile (b) Acceleration profile

Fig. 9. Velocity and acceleration of the 1st joint.

simulation platform can calculate torques with more precise mass distribution, while the Matlab solver calculates the mass matrices using simple and regular geometry of links. The difference in mass matrices is demonstrated in Eq. (17) for Link 2, where $\mathbf{I}'_2$ is calculated by the Matlab solver and $\mathbf{I}_2$ is by ADAMS.

Table 2
Candidate motor data from Maxon Motor [26].

Index no.	Maxon Motor	T_m (Nm)	T_m^{max} (Nm)	N_m^{max} (rpm)	J_m (g cm^2)	m_m (kg)
1	EC 45 flat	0.0843	0.822	10,000	135	0.11
2	RE 25	0.0284	0.28	14,000	10.5	0.13
3	RE 26	0.0321	0.227	14,000	12.1	0.15
4	EC-i 40	0.0667	1.81	15,000	24.2	0.21
5	RE 30	0.0882	1.02	12,000	34.5	0.238
6	EC 32	0.0426	0.353	25,000	20	0.27
7	RE 35	0.0965	0.967	12,000	67.4	0.34
8	RE 36	0.0795	0.785	12,000	67.2	0.35
9	EC 40	0.127	0.94	18,000	85	0.39
10	RE 40	0.184	2.5	12,000	138	0.48

Table 3
Candidate gearbox data from harmonic drive [27].

Index no.	CPU unit size	Ratio	T_g (Nm)	T_g^{max} (Nm)	N_g^{max} (rpm)	J_g (kg m^2)	m_g (kg)
1	14	100	11	54	8500	0.033×10^{-4}	0.54
2	17	100	39	110	7300	0.079×10^{-4}	0.79
3	20	100	49	147	6500	0.193×10^{-4}	1.3
4	25	100	108	284	5600	0.413×10^{-4}	1.95

Table 4
Optimization results for minimization of weight.

Joint	Initial		Optimized	
	Motor	Gearbox	Motor	Gearbox
1	RE 40	CPU 17	EC 40	CPU 17
2	RE 35	CPU 17	EC 45 flat	CPU 17
3	RE 35	CPU 17	EC-i 40	CPU 14
4	RE 35	Gearhead	EC 45 flat	Gearhead
5	RE 35	CPU 17	EC 45 flat	CPU 14
Arm weight (kg)	16.7		10.2	

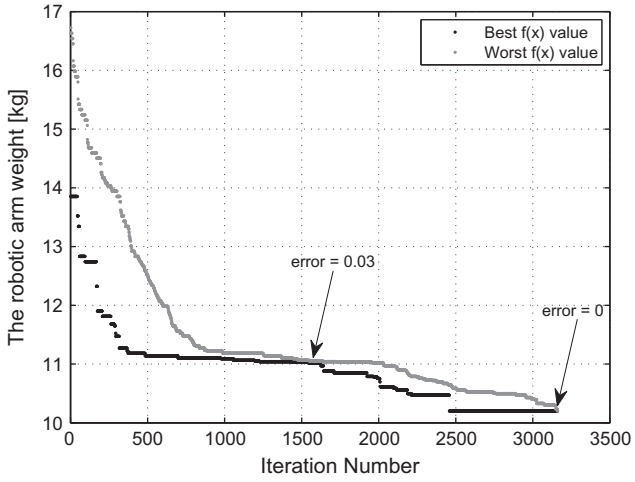


Fig. 10. Convergence plot for the weight of the robotic arm.

$$\mathbf{I}_2' = \begin{bmatrix} 0.0076 & 0 & 0 \\ 0 & 0.0076 & 0 \\ 0 & 0 & 0.0045 \end{bmatrix} (\text{kg m}^2),$$
$$\mathbf{I}_2 = \begin{bmatrix} 0.008 & -0.00032 & -0.00008 \\ -0.00032 & 0.0069 & 0.00037 \\ -0.00008 & 0.00037 & 0.0049 \end{bmatrix} (\text{kg m}^2) \quad (17)$$

6.2. Design variables programming

The design points in the Complex method are usually continuous. But the design variables $u_{m,i}$ and $u_{g,i}$ have to be integers, since they are the index numbers from the databases of motors and gearboxes. Two ways of dealing with the design variables are investigated in order to confirm a more efficient one. One way is called *rounded design variable (RDV)*, the other one called *linear design variable (LDV)*.

Rounded design variable (RDV). For the RDV method, a rounding function is introduced to transfer the design variables into integers. The rounding function is given as

$$x_{DV} = round(x) = \begin{cases} x_{int}; & \text{if } x_{int} \leqslant x < x_{int} + 0.5 \\ x_{int} + 1; & \text{if } x_{int} + 0.5 \leqslant x < x_{int} + 1 \end{cases} \quad (18)$$

where x is the design variable manipulated by the Complex method, x_{int} is the integral part of the number x, and x_{DV} is the rounded design variable. x_{DV} is used to update the mass of motors and gearboxes in inverse dynamic analysis, as well as the allowable torque and speed values used to examine constraint violations.

Linear design variable (LDV). For the LDV method, the mass between two adjacent motors (or gearboxes) in the category is linearized by the function

$$m(x) = m(x_{int}) + (x - x_{int}) \cdot [m(x_{int} + 1) - m(x_{int})] \quad (19)$$

where $m(x_{int})$ is the mass of the component (motors from Table 2 and gearboxes from Table 3) corresponding to the index number x_{int},$m(x)$ is the mass to be updated for the component in inverse dynamic analysis.

Comparison has been conducted upon the RDV and LDV methods of dealing with the integer design variables, as listed in Table 5. In general, the LDV method yields better results at the cost of more iterations and objective function evaluations. In the above example, the RDV method has been preferred, however, the choice of method must in general be a compromise between accuracy and optimization time.

7. Conclusions

A method was developed for the optimum design of robotic drive trains. The selection of motors and gears was formulated as a discrete optimization problem, which was solved by a non-gradient optimization algorithm. Constraints were formulated by considering both motor and gearbox characteristics and robotic arm dynamics. The proposed method is able to reach a design with lower mass for a given set of driving components. A co-simulation platform consisting of a MSC.ADAMS dynamics model and an optimization algorithm implemented in Matlab code was developed, which enables design optimization based on dynamics of an embodiment existing in CAD systems. Such a platform not only yields accurate dynamic calculation for drive train optimization, but also leads to a possible integrated optimization for both arm structures and drive trains.

Acknowledgements

The project is sponsored by Aalborg University and Det Obelske Familiefond (DOF). Funding from the Chinese Scholarship Council (CSC) to Lelai Zhou is gratefully acknowledged.

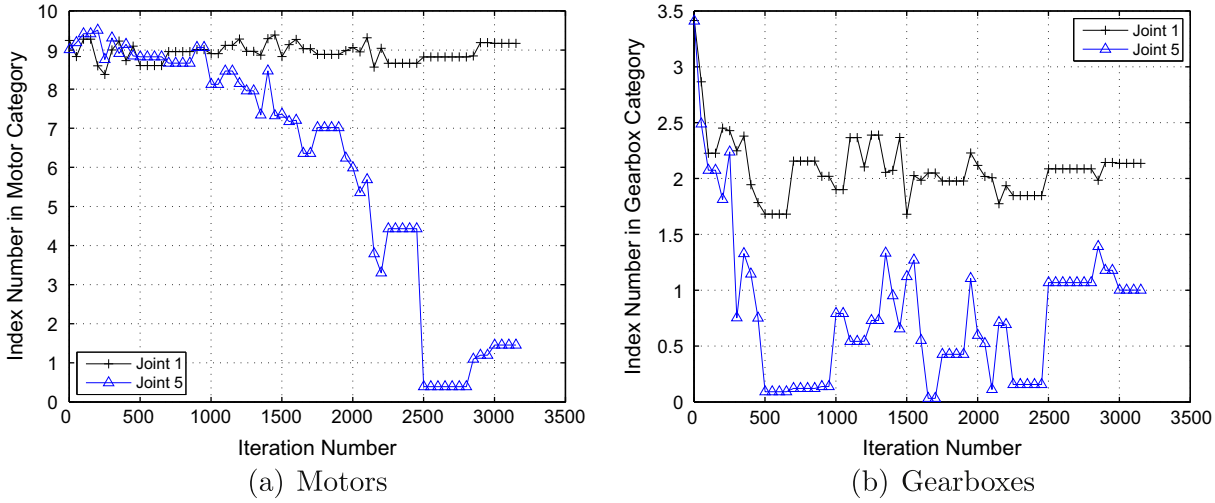


Fig. 11. Convergence plots for design variables.

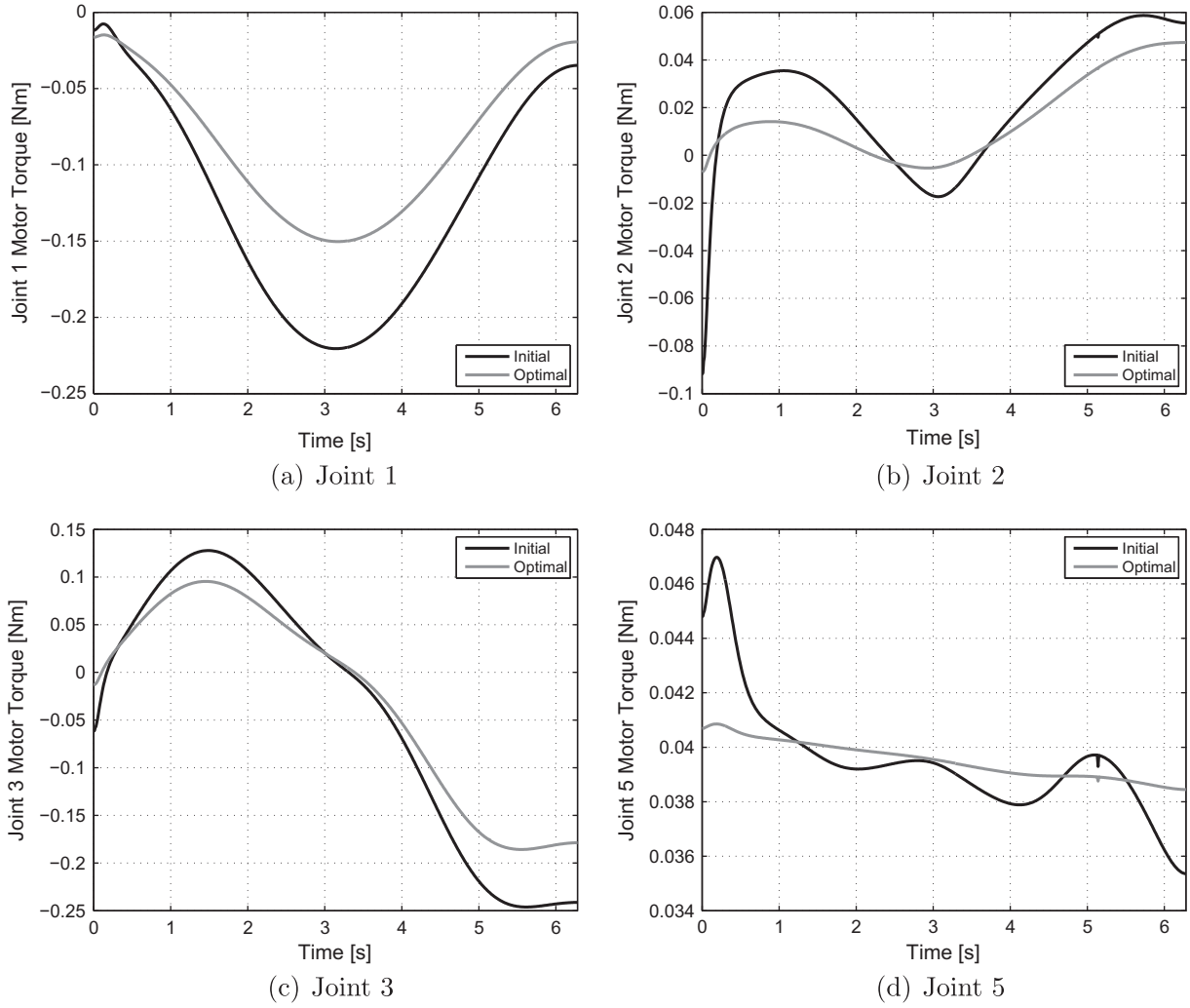


Fig. 12. Motor torques for initial and optimal drive train combinations.

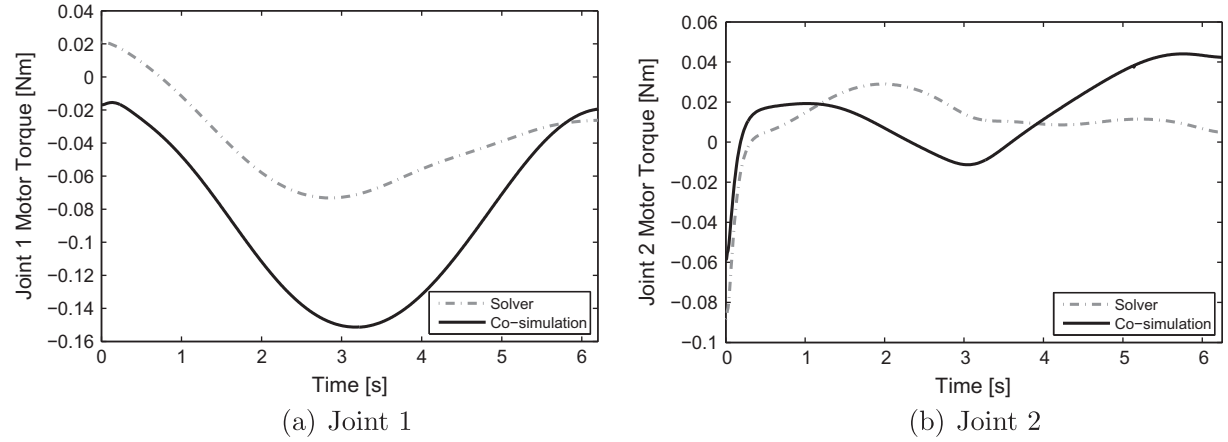


Fig. 13. Comparisons of joint torques solved by the co-simulation platform and a Matlab solver.

Table 5
The comparisons of RDV and LDV.

Population	RDV		LDV	
	$f(\mathbf{x})$	Iteration	$f(\mathbf{x})$	Iteration
20	11.142	213	10.633	1499
40	10.952	927	10.703	2607
60	10.458	1444	10.414	6303
80	10.402	3026	10.122	7306

Appendix A

The transformation matrix in forward kinematics of the end-effector in fixed reference frame is given as:

$$ {}^{0}A_5 = \begin{bmatrix} \mathbf{R} & \mathbf{q} \\ 0 & 1 \end{bmatrix} \tag{A.1} $$

with the rotation matrix **R** and position vector **q** being given by

$$\mathbf{R} = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} \tag{A.2}$$

The forward kinematics of the robotic arm is solved as below:

$$\begin{aligned}
u_x &= c\theta_5(s\theta_1 s\theta_4 + c\theta_1 c\theta_{23} c\theta_4) - c\theta_1 s\theta_{23} s\theta_5,\\
u_y &= -c\theta_5(c\theta_1 s\theta_4 - s\theta_1 c\theta_{23} c\theta_4) - s\theta_1 s\theta_{23} s\theta_5,\\
u_z &= c\theta_{23} s\theta_5 + s\theta_{23} c\theta_4 c\theta_5,\\
v_x &= s\theta_1 c\theta_4 - c\theta_1 c\theta_{23} s\theta_4,\\
v_y &= -s\theta_1 c\theta_{23} s\theta_4 - c\theta_1 c\theta_4,\\
v_z &= -s\theta_{23} s\theta_4,\\
w_x &= s\theta_5(s\theta_1 s\theta_4 + c\theta_1 c\theta_{23} c\theta_4) + c\theta_1 s\theta_{23} c\theta_5,\\
w_y &= -s\theta_5(c\theta_1 s\theta_4 - s\theta_1 c\theta_{23} c\theta_4) + s\theta_1 s\theta_{23} c\theta_5,\\
w_z &= s\theta_{23} c\theta_4 s\theta_5 - c\theta_{23} c\theta_5,\\
q_x &= l_1 c\theta_1 c\theta_2 + l_2 c\theta_1 s\theta_{23} + d_1 \cdot (s\theta_1 c\theta_4 - c\theta_1 c\theta_{23} s\theta_4),\\
q_y &= l_1 s\theta_1 c\theta_2 + l_2 s\theta_1 s\theta_{23} - d_1 \cdot (c\theta_1 c\theta_4 + s\theta_1 c\theta_{23} s\theta_4),\\
q_z &= h_1 + l_1 s\theta_2 - l_2 c\theta_{23} - d_1 s\theta_{23} s\theta_4,
\end{aligned}$$

where $c\theta_{23}$ stands for $\cos(\theta_2 + \theta_3)$, and $s\theta_{23}$ for $\sin(\theta_2 + \theta_3)$.

Appendix B

The joint angles for a given pose in terms of **R** and **q** can be found through the inverse kinematics presented below.

Skipping details, a solution for θ_1 is found as

$$\theta_1 = \arctan\left(\frac{p_y}{p_x}\right) \tag{B.1}$$

where $p_x = q_x - d_1 w_x$ and $p_y = q_y - d_1 w_y$. Eq. (B.1) leads to two solutions of θ_1, i.e. $\theta_1 = \theta_1^*$ and $\theta_1 = \theta_1^* + \pi$, where $0 \leqslant \theta_1^* \leqslant \pi$. They represent two branches of arm kinematics. The real value depends on the initial configuration.

The solution of θ_3 is given as

$$\theta_3 = \arctan\left(\pm\frac{\kappa_3 - \kappa_1}{\sqrt{\kappa_2^2 - (\kappa_3 - \kappa_1)^2}}\right); \quad \kappa_2^2 - (\kappa_3 - \kappa_1)^2 > 0 \tag{B.2}$$

where $\kappa_1 = l_1^2 + l_2^2, \kappa_2 = 2l_1 l_2, \kappa_3 = p_x^2 + p_y^2 + (p_z - h_1)^2$, and $p_z = q_z - d_1 w_z$.

Once θ_1 and θ_3 are known, θ_2 can be obtained as

$$\theta_2 = \arctan\frac{(\mu_2\eta_1 - \mu_1\eta_2)(\zeta_2\mu_1 - \zeta_1\mu_2)}{(\mu_2\zeta_1 - \mu_1\zeta_2)(\zeta_2\eta_1 - \zeta_1\eta_2)} \tag{B.3}$$

where

$$\begin{aligned}
&\mu_1 = l_1 + l_2 s\theta_3, \ \zeta_1 = l_2 c\theta_3, \ \eta_1 = p_x c\theta_1 + p_y s\theta_1;\\
&\mu_2 = -l_2 c\theta_3, \ \zeta_2 = l_1 + l_2 s\theta_3, \ \eta_2 = p_z - h_1.
\end{aligned}$$

θ_5 takes the form of

$$\theta_5 = \arccos(w_x c\theta_1 s\theta_{23} + w_y s\theta_1 s\theta_{23} - w_z c\theta_{23}) \tag{B.4}$$

Assuming that $s\theta_5 \neq 0$, we can solve for θ_4 as follows.

$$\theta_4 = \arctan(s\theta_4, c\theta_4) \tag{B.5}$$

where

$$c\theta_4 = \frac{w_x c\theta_1 c\theta_{23} + w_y s\theta_1 c\theta_{23} + w_z s\theta_{23}}{s\theta_5}; \ s\theta_4 = \frac{w_x s\theta_1 - w_y c\theta_1}{s\theta_5}.$$

References

[1] Pasch K, Seering W. On the drive systems for high-performance machines. ASME J Mech 1983;106:102–8.
[2] van De Straete HJ, Schutter JD, Degezelle P, Belmans R. Servo motor selection criterion for mechatronic applications. IEEE Trans Mech 1998;3(1):43–50.
[3] van De Straete HJ, Schutter JD, Belmans R. An efficient procedure for checking performance limits in servo drive selection and optimization. IEEE Trans Mech 1999;4(4):378–86.
[4] Cetinkunt S. Optimal design issues in high-speed high-precision motion servo systems. Mechatronics 1991;1:187–201.
[5] Cusimano G. A procedure for a suitable selection of laws of motion and electric drive systems under inertial loads. Mech Mach Theory 2003;38:519–53.
[6] Cusimano G. Optimization of the choice of the system electric drive-device-transmission for mechatronic applications. Mech Mach Theory 2007;42:48–65.
[7] Roos F, Johansson H, Wikander J. Optimal selection of motor and gearhead in mechatronic applications. Mechatronics 2006(16):63–72.
[8] Chedmail P, Gautier M. Optimum choice of robot actuators. J Eng Ind 1990;112:361–7.
[9] Pettersson M, Ölvander J. Drive train optimization for industrial robots. IEEE Trans Robotics 2009;25(6):1419–23.
[10] Leger C. Automated synthesis and optimization of robot configurations: an evolutionary approach. Ph.D. thesis. Pittsburgh (USA): Carnegie Mellon University; 1999.
[11] Elmqvist H, Olsson H, Mattsson SE, Brück D. Optimization for design and parameter estimation. In: Proceedings of the 4th international Modelica conference; 2005. p. 255–66.
[12] Albu-Schäffer A, Haddadin S, Ott C, Stemmer A, Wimböck T, Hirzinger G. The DLR lightweight robot: design and control concepts for robots in human environments. Ind Robot 2007;34(5):376–85.
[13] Izumi T, Zhou H, Li Z. Optimal design of gear ratios and offset for energy conservation of an articulated manipulator. IEEE Trans Automat Sci Eng 2009;6(3):551–7.
[14] Chen D-Z. Drive train configuration arrangement for gear coupled manipulators. J Robot Syst 1997;14(8):601–12.
[15] Chen D-Z, Wang S-C. On the drive train design of gear coupled manipulators. J Robot Syst 1998;15(8):477–86.
[16] Box MJ. A new method of constrained optimization and a comparison with other methods. Comput J 1965(8):42–52.
[17] Zhou L, Bai S, Hansen MR. Design and kinematics of a 5-dof light-weight anthropomorphic robotic arm. In: Proceedings of the 22nd Nordic seminar on computational mechanics, no. 11. Aalborg, Denmark; 2009. p. 205–8.
[18] Denavit J, Hartenberg RS. A kinematic notation for lower pair mechanisms based on matrices. ASME J Appl Mech 1955;77:215–21.
[19] Tsai LW. Robot analysis: the mechanics of serial and parallel manipulators. John Wiley & Sons; 1999.
[20] Angeles J. Fundamentals of robotic mechanical systems: theory, methods, and algorithms. 3rd edition. Springer; 2007.
[21] Hollerbach JM. A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. IEEE Trans Syst, Man, Cybernet SMC-10 1980(11):730–6.
[22] Luh JYS, Walker MW, Paul RP. On-line computational scheme for mechanical manipulators. ASME J Dyn Syst, Meas, Control 1980;120:69–76.
[23] Engineering data for harmonic drive gears. <www.harmonicdrive.de/cms/upload/pdf/en/cpu_h7.pdf>.
[24] Norten RL. Machine design: an integrated approach. 4th edition. Prentics Hall; 2010.
[25] Antony GG. Rating and sizing of precision low backlash planetary gearboxes for automation motion control and robotics applications. <www.neugartusa.com/Service/faq/Gear_Rating.pdf>.
[26] Maxon Motor products catalogue 10/11. <www.maxonmotor.ch/e-paper/blaetterkatalog/pdf/complete.pdf>.
[27] Harmonic Drive technical data. <www.harmonicdrive.de/cms/upload/German/B_Produkte/B_Units/kompl_Produktkapitel_CPU_D-E.pdf>.