

Cost-Optimal Composition Synthesis for Modular Robots

Esra Icer and Matthias Althoff

Abstract—The ongoing trend of increasing product individualization requires more flexible solutions in production systems. Modular robots address this demand since they can be assembled in different ways from a given set of modules. One of the reasons why modular robots are not yet successfully introduced in the market is that it is not clear how to assemble modules such that the robot will be able to achieve a specific task optimally, especially in the presence of obstacles. This problem is challenging since a huge combination of possible assemblies exists and one has to find the optimal trajectory for each of them. We address exactly this issue not by finding optimal solutions for each assembly, but instead pruning the search space: First, we remove assemblies that cannot achieve the task before starting the process of finding optimal trajectories. Second, we exploit the iterative nature of numerical optimization routines by removing assemblies that are not promising in each iteration. We demonstrate that our approach is clearly better compared to finding assemblies by optimizing trajectories for each assembly individually.

I. INTRODUCTION

Adapting industrial robots to changing tasks or environments is difficult although they are widely used due to their efficiency, robustness, and accuracy. Modular robot manipulators consisting of several interchangeable, pre-designed joint modules and link modules which enable us to configure various manipulators for different tasks from a set of given modules are designed as a solution to this problem. The unique properties of modular robots such as flexibility, easy modification, easy maintenance and high versatility make them a promising technology for future flexible manufacturing scenarios. One of the main challenges facing modular robots is the problem of determining the optimal composition of given modules for given tasks. We consider that the task is to carry a given payload from an initial pose to a final pose in a cost-optimal way without colliding with any obstacles in the environment. In this work, we propose a time-efficient algorithm for selecting the optimal module combination from a set of given modules considering kinematics and dynamics so that the predefined task is fulfilled.

Selecting the optimal composition of modular robot manipulators has been researched for the last 30 years. We review previous works on task-based configuration synthesis of modular robots and the selection of the optimal configuration among the possible compositions considering the desired objective function. Assembly synthesis methods for modular robots which are based on assembly matrices that have the information about the module type, module length and connection type are presented in [1]–[3]. A genetic algorithm

(GA) based synthesis method for modular robots to obtain the optimal composition for a given task regarding modules' physical properties is proposed in [4]. A two-level synthesis method for non-redundant manipulators consisting of links with adjustable lengths and rotational joints is presented in [5]. As a first step of that work, the kinematic composition of the robot is determined considering the given task and then a GA is used to obtain the optimal lengths of the link modules for the pre-defined composition [5]. Another two-level GA-based method to obtain the optimal composition at the given discrete poses is introduced in [6]. The top level GA generates an assembly from the given modules and the lower level GA optimizes joint positions for the task in an environment with static obstacles. A GA-based modular robot composition synthesis method which obtains optimal kinematic configurations for task-based, fault-tolerant manipulators is proposed in [7]. In [8], a simulated annealing-based optimal composition synthesis method only regarding the kinematic constraints is presented. A concurrent optimal design approach, which mainly depends on grouping the variables to reduce the number of independent variables in the optimization process, is proposed in [9]. A progressive task-based design approach for non-modular manipulators is proposed in [10], which divides the design procedure into kinematic design, planning, and kinematic control. All previously mentioned works penalize unfeasible compositions during the optimization process only considering the initial and the goal positions. In our previous work [11], we propose a composition synthesis method based on the elimination of the unfeasible compositions not only taking into account the initial and the goal feasible positions but also considering if the given task is achievable or not. All the aforementioned methods consider only the kinematics to obtain the optimal solution.

To optimally solve the predefined task, the dynamics of modular robots should also be taken into account. However, classical dynamic model generation techniques are for fixed robot structures and it is not efficient to generate the dynamic model for each individual composition. As an affordable solution to this problem, an automated design method considering the kinematic and the dynamic model for modular robots, which is also applicable to parallel robots, is presented in [12]. Then, given joint modules are assembled regarding the given task specifications and then link modules are designed for the previously defined assembly. Another technique to automatically generate dynamic models of modular robots based on the theory of Lie algebras is proposed in [13]. In [14], all kinematic and dynamic parameters are stored in each module and the dynamics of the desired composition is

The authors are with the Department of Computer Science, Technische Universität München, 85748 Garching, Germany. Corresponding e-mail: {icer, althoff}@in.tum.de

generated automatically. In this paper, we use the Newton-Euler method as in [14] to generate the dynamics because of its simplicity.

In addition to the composition synthesis method and kinematic and dynamic model generation, determination of how to achieve the given task is also a problem for modular manipulators. Since there is a large number of different manipulators to consider, a fast trajectory planning algorithm is required to achieve the given task. There are two main approaches for optimal collision-free trajectory planning problems: *i*) obtaining the cost-optimal trajectory following the pre-defined path [15], [16], and *ii*) using optimal control methods considering the initial and the goal positions [17], [18]. The methods in the first category may fail to find the optimal path and non-collision is not guaranteed because of the possibility of collision between pre-determined positions. Direct methods for trajectory optimization, which are in the second category, are based on the transformation of the optimal control problem into a nonlinear programming (NLP) problem. They are detailed in [17] and it is shown that the multiple direct shooting method is one of the best methods to obtain the optimal trajectory for a robot arm considering the obstacles in the environment. Another direct optimal control method is used in [18] to obtain the time-optimal trajectory for manipulators considering collision avoidance from static and dynamic obstacles as state constraints. All previously mentioned methods have been applied to fixed-structure robots. In this paper, we introduce an algorithm to obtain the optimal composition for a given task considering a set of given modules. To the best knowledge of the authors, this is the first work which determines the optimal composition considering how to achieve the given task, taking into account the kinematic and the dynamic constraints. Our proposed approach differs from the previously proposed methods in *i*) we automatically generate kinematic model and dynamic model for feasible compositions, *ii*) we find an individual solution for each feasible composition considering kinematic and dynamic task requirements, and *iii*) we propose a composition elimination method which is mainly based on the elimination of less-likely compositions.

This paper is organized as follows: Sec. II defines the combinatorial configuration synthesis problem for modular robots. Sec. III explains our configuration synthesis method, followed by implementation details in Sec. IV. Finally, conclusions are drawn in Sec. V.

II. PROBLEM STATEMENT

We consider reconfigurable, serially-connected modular robot manipulators with n degrees of freedom (DOFs), the kinematics of which are determined uniquely by a vector $\mathbf{q} \in \mathbb{R}^n$ of joint positions where angles stand for rotational joints and translations stand for prismatic joints. Throughout this paper, we assume that *i*) we consider four different types of modules which are bases, joints, links and end-effectors, *ii*) all modules have only one input and one output connection port, and *iii*) all manipulators begin with a base module, end

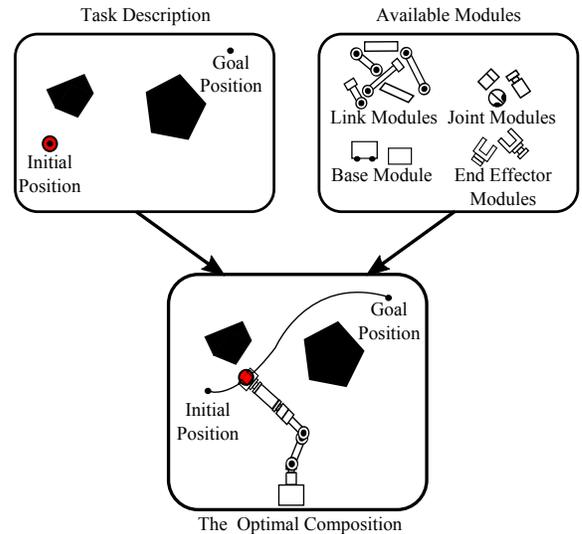


Fig. 1. The problem of finding the optimal composition for task-based modular robots.

with an end effector module and there are only joint and link modules between the base and end effector modules. These assumptions are fairly realistic for robots used in industry for different scenarios. Without loss of generality, the variable $k \in \{1, \dots, N\}$ uniquely refers to a possible composition and the variable N refers to the maximum number of possible compositions without considering task-defined constraints. The task requirements are constrained by the kinematic model of the robot composition, the dynamic model of the robot composition and static obstacles in the environment. The space of the environment, which contains the robot and obstacles, is denoted by $\mathcal{W} \subset \mathbb{R}^3$ and we call the subset of space occupied by the robot as $\mathcal{A} \subset \mathcal{W}$. The occupancy of the robot for a specific joint position vector \mathbf{q} is indicated by $\mathcal{A}(\mathbf{q}) \subset \mathcal{W}$ and the variable $\mathcal{A}(\mathbf{q}_k) \subset \mathcal{W}$ refers to the space occupied by the k^{th} composition. Arbitrary geometric shapes represent the obstacles in \mathbb{R}^3 and the union of all obstacles is denoted by $\mathcal{O} = \bigcup_j \mathcal{O}_j$. For the sake of simplicity, each obstacle in the environment is considered to be enclosed with a sphere. Obstacle-free space in the environment is defined as $\mathcal{F} = \mathcal{W} \setminus \mathcal{O}$. The time variable is $t \in [0, t_f]$, where t_f is the final time to reach the goal. The vector $\mathbf{q}(t)$ maps the time t to the joint position vector; initial time is assumed to be zero. The forward kinematic function from the joint position vector $\mathbf{q}(t)$ to the end effector position is denoted by $f(\mathbf{q}(t))$. We define the all path planning problems starting from a given initial position \mathbf{p}_s , defined as $\mathbf{p}_s = f(\mathbf{q}(0))$, and end at a given position \mathbf{p}_g , defined as $\mathbf{p}_g = f(\mathbf{q}(t_f))$.

The dynamic model of the modular robot is obtained from (1), where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the symmetric and definite mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ is the Coriolis and centrifugal matrix, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is the vector of gravity term, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of actuation forces/torques, $\dot{\mathbf{q}}(t)$ is the vector of joint velocity and $\ddot{\mathbf{q}}(t)$ is the vector of joint acceleration (see

[19]).

$$\boldsymbol{\tau}(t) = \mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \dot{\mathbf{q}}(t) + \mathbf{g}(\mathbf{q}(t)) \quad (1)$$

The set of module combinations, \mathcal{K} , that fulfill a given task, is described as:

$$\begin{aligned} \mathcal{K} = \{k \mid & 1 \leq k \leq N \wedge \forall t \in [0, t_{f,k}] \exists \mathbf{q}_k(\cdot) : \\ & \wedge \mathcal{A}(\mathbf{q}_k(t)) \in \mathcal{F} \\ & \wedge \mathbf{q}_k(t) \in [\mathbf{q}_{k,min}, \mathbf{q}_{k,max}] \\ & \wedge \dot{\mathbf{q}}_k(t) \in [\dot{\mathbf{q}}_{k,min}, \dot{\mathbf{q}}_{k,max}] \\ & \wedge |\boldsymbol{\tau}_k(\mathbf{q}_k(t))| \leq \boldsymbol{\tau}_{k,max} \\ & \wedge f_k(\mathbf{q}_k(0)) = \mathbf{p}_s \wedge f_k(\mathbf{q}_k(t_{f,k})) = \mathbf{p}_g \}. \end{aligned} \quad (2)$$

The optimal composition, which fulfills the given task considering the minimum cost, is obtained from (3). The composition that gives the cost-optimal solution is denoted by κ and the optimal cost value for the k^{th} composition is denoted by c_k :

$$\kappa = \underset{k \in \mathcal{K}}{\text{argmin}} c_k. \quad (3)$$

III. PROPOSED METHOD

Increasing the number of the modules and changing their assemblies, exponentially increases the design space of possible compositions, which makes the task-based configuration synthesis problem complex and computationally expensive. In our previous work [11], we propose a composition synthesis method which aims to reduce the computational time by eliminating unfeasible compositions via required tests to achieve the given task. These tests go from simpler ones to more complicated and time-consuming ones and only consider kinematic task requirements. In addition to our previous work, we consider dynamics to determine the cost-optimal composition in this paper. Throughout the paper, the task is defined as carrying the pre-defined payload from a given initial position to a given goal position in a cost-optimal way without colliding with any obstacles in the environment (see Fig. 1). The schematic representation of our proposed method is shown in Fig. 2, which consists of five steps: (A) generation of all possible compositions, (B) generation of feasible compositions considering the initial and the goal positions, (C) generation of the cost-optimal solutions for the remaining compositions from (B), (D) elimination of the cost-inefficient compositions in each iteration of the optimization process

process, and (E) selection of the cost-optimal composition. The first two steps of the proposed method, (A) and (B), are detailed in [11]. For the following steps, the robot dynamics is considered and compositions which are unlikely to be the optimal solution are eliminated to reduce the computational time while obtaining the optimal composition regarding the desired objective function.

(C) Generation of the Cost-Optimal Trajectories

The general form of our optimal control problem for ordinary differential equations (ODEs) in the Lagrangian form is defined as in (4), where $\mathbf{x}(\cdot)$ is the vector of state variables, defined as $\mathbf{x} = (\mathbf{q}; \dot{\mathbf{q}})$ where $\mathbf{x} \in \mathbb{R}^{2n}$, $\mathbf{u}(\cdot)$ is the vector of control variables which are defined as the torques/forces acting on the joints, where $\mathbf{u} = \boldsymbol{\tau}$ and $\mathbf{u} \in \mathbb{R}^n$, and $L(\mathbf{x}(t), \mathbf{u}(t))$ is the desired objective function as in [17]. The robot dynamics is transformed from (4) to the state space form $\dot{\mathbf{x}} = g(\mathbf{x}_k(t), \mathbf{u}_k(t))$. Point to point motion is constrained by $\mathbf{x}(0) = \mathbf{x}_0$ and $\mathbf{x}(t_f) = \mathbf{x}_f$ where \mathbf{x}_0 and \mathbf{x}_f indicate the initial and the final state variables obtained from (B), respectively. The joint limits and collision avoidance constraints are defined as inequality constraints in the function $h(\mathbf{x}(t), \mathbf{u}(t))$.

$$\underset{\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f}{\text{minimize}} \int_0^{t_f} L(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (4)$$

subject to

$$\begin{aligned} \mathbf{x}(0) - \mathbf{x}_0 &= 0, & \text{(initial constraints)} \\ \dot{\mathbf{x}}(t) - g(\mathbf{x}(t), \mathbf{u}(t)) &= 0, & \text{(ODE model)} \\ h(\mathbf{x}(t), \mathbf{u}(t)) &\geq 0, & \text{(path constraints)} \\ \mathbf{x}(t_f) - \mathbf{x}_f &= 0. & \text{(terminal constraints)} \end{aligned}$$

Some of the most used trajectory planning methods are direct methods which convert the infinite optimal control problem in (4) into a finite dimensional NLP problem [20]. When compared to other methods, direct methods have a good balance of computational efficiency and accuracy [17]. As a solution to our problem, we use the direct multiple shooting method due to its robustness, fast convergence, easy parallelizability and applicability for unstable systems [17]. After obtaining the set of feasible compositions at the initial and the goal positions, we implement the direct multiple shooting method for finding optimal compositions, which passed the tests in (B). For the direct multiple shooting

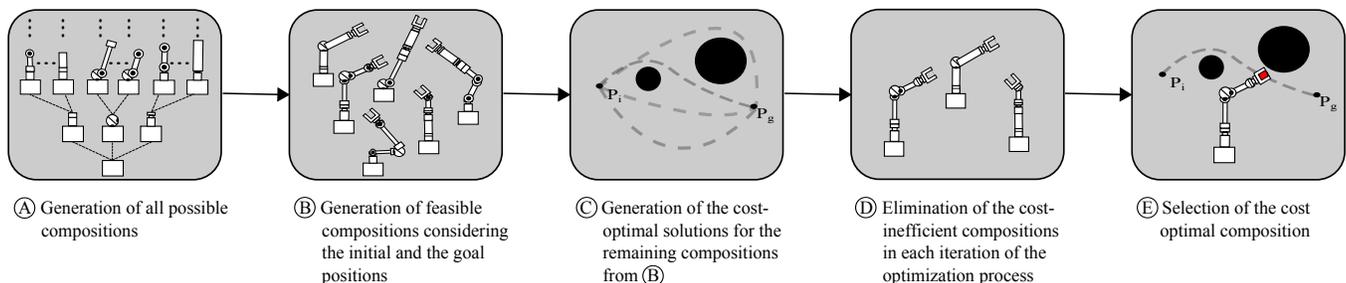


Fig. 2. The cost-optimal composition synthesis procedure.

method, the state variables and the control variables are discretized into N_{int} pieces between the initial and the goal positions. To simplify the formula, each discretized control value is defined as $\{\forall t : \mathbf{u}(t_{int}) = \mathbf{b}_m, t_{int} \in [t_m, t_{m+1}]\}$. The variable m is defined as $m \in \{0, \dots, N_{int}-1\}$. The ODE solver is individually used in each t_{int} and it is considered that each interval starts with an artificial value \mathbf{s}_m , which is defined as $\mathbf{x}_m(t_m) = \mathbf{s}_m$. The solutions of the ODE solver are trajectory pieces $\mathbf{x}_m(t_m; \mathbf{s}_m, \mathbf{b}_m)$, where $\mathbf{s}_{m+1} = \mathbf{x}_m(t_{m+1}; \mathbf{s}_m, \mathbf{b}_m)$ must be satisfied for all times.

We reformulate the optimal control problem for modular robots in (5) to obtain the cost-optimal trajectory for the k^{th} composition with the following definitions: $l_{k,m}$ is the discretized objective function for the k^{th} composition which is in the interval $[t_{k,m}, t_{k,m+1}]$. The variables $\mathbf{x}_{k,max}$ and $\mathbf{x}_{k,min}$ are the maximum and the minimum state variables for the k^{th} composition, $\mathbf{u}_{k,max}$ and $\mathbf{u}_{k,min}$ are the maximum and the minimum control variables for the k^{th} composition and r_j is the radius of the j^{th} obstacle. Furthermore, the occupancy set of the robot is projected onto a function $d_k(t)$, which represents the distance between the robot components and the j^{th} obstacle at time t .

$$\underset{\mathbf{s}_k, \mathbf{b}_k}{\text{minimize}} \sum_{m=0}^{N_{int}-1} l_{k,m}(\mathbf{s}_{k,m}, \mathbf{b}_{k,m}) \quad (5)$$

subject to

$$\begin{aligned} \mathbf{s}_{k,0} - \mathbf{x}_{k,0} &= 0, & (\text{initial constraints}) \\ \mathbf{s}_{k,m+1} - \mathbf{x}_{k,m}(t_{k,m+1}; \mathbf{s}_{k,m}, \mathbf{b}_{k,m}) &= 0, & (\text{ODE model}) \\ h(\mathbf{s}_{k,m}, \mathbf{b}_{k,m}) &\geq 0, & (\text{path constraints}) \\ \mathbf{s}_{k,N_{int}} - \mathbf{x}_{k,f} &= 0 & (\text{terminal constraints}) \end{aligned}$$

where

$$h(\mathbf{s}_{k,m}, \mathbf{b}_{k,m}) = \begin{pmatrix} \mathbf{x}_{k,max} - \mathbf{s}_{k,m} \\ \mathbf{s}_{k,m} - \mathbf{x}_{k,min} \\ \mathbf{u}_{k,max} - \mathbf{b}_{k,m} \\ \mathbf{b}_{k,m} - \mathbf{u}_{k,min} \\ d_k(t) - r_j \end{pmatrix}.$$

Defining the optimal trajectory, collision avoidance is considered as an inequality constraint in the optimization problem. We apply task space collision avoidance algorithms due to the complexity of mapping the environment onto each composition's configuration space. To implement the task space collision avoidance, task space coordinates of each module are obtained for each time interval. This process is realized by using the transformation matrices of the Denavit and Hartenberg (D-H) frames and an additional translation for the i^{th} link which is defined as n_i in [14]. All link modules are modeled as lines, and base module, all joint modules and end effector modules are modeled as points. The thickness of each module is added to the radii of the obstacles while checking the collisions between the

modules and the obstacles in the environment. Although this assumption provides larger obstacles than real ones, it dramatically reduces the computational time when compared to other collision checking methods.

Ⓓ Elimination of the Cost-Inefficient Compositions in Each Iteration

Implementation of the optimal control method to obtain the cost-optimal trajectories for all compositions, which fulfill the tests in Ⓒ, is time-consuming. As a solution to this problem, we propose a composition elimination method for modular robots. Based on the proposed method, all feasible compositions are considered for the initial iterations and the multiple direct shooting method is applied to all of them. We roughly optimize all feasible compositions in parallel constraining the maximum number of iterations and record the obtained cost values in each iteration. Starting from the second iteration, we compare the change in cost values for each composition with the change in cost values for the other compositions with the formula $\frac{c_{k,l} - c_{min,l}}{c_{k,l}}$, where $c_{k,l}$ is the cost value of the k^{th} composition in the l^{th} iteration and $c_{min,l}$ is the minimum cost value among all compositions in the l^{th} iteration. In case the obtained difference rate is greater than a specific value, we discard the composition. We repeat this procedure for every composition in each iteration and thus eliminate the compositions less likely to be the optimal until we find the optimal cost value for the remaining compositions.

Ⓔ Selection of the Cost-Optimal Composition

To obtain the cost-optimal module composition, we compare the results remaining from the previous section. After the optimization procedure ends, we may have several remaining possible solutions. To find the optimal composition, we consider the cost values and select the composition with the minimum cost value as the optimal solution as shown in (3).

IV. NUMERICAL EXPERIMENTS

To demonstrate the applicability of our proposed algorithm, we implement it in MATLAB R2016a on an Intel® Core™ i7 processor with 2.30 GHz and 16 GB of memory. We use one type of fixed base module, two types of one DOF joint modules (rotational (R) and prismatic (P)), three types of zero DOF link modules ($\alpha_1 = 90^\circ$, $\alpha_2 = 0^\circ$, and $\alpha_3 = 90^\circ$) and two types of one DOF end effector modules (rotational (R) and prismatic (P)), see Fig. 3. The required properties for each module are given in Tab. I. For all compositions, we consider serially-connected modular manipulators whose bases are located at $(0, 0, 0)^T$. The task is defined as carrying the given payload from the given initial position to the given goal position in the shortest time without colliding with the obstacles in the environment and without violating the joint limits with the maximum 5-DOFs robot. The payload is given as 50 kg, the initial and the goal positions are defined as $\mathbf{p}_s = (1.5, 1.5, 2)^T$ and $\mathbf{p}_g = (-2, -1.2, 0.6)^T$, respectively. It is assumed that

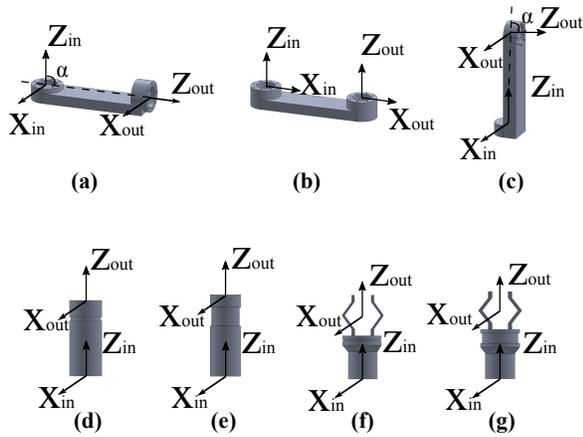


Fig. 3. Basic modules used in this study, (a) $\alpha = 90^\circ$ link module, whose length is along the y direction of the previous coordinate system (L1), (b) $\alpha = 0^\circ$ link module (L2), (c) $\alpha = -90^\circ$ link module, whose length is along z direction of the previous coordinate system (L3), (d) rotational joint module (J1), (e) prismatic joint module (J2), (f) rotational end effector module (E1), (g) prismatic end effector module (E2)

there are two static obstacles in the environment whose center points are defined as $\mathbf{p}_{1,O} = (1.5, -1.3, 1.1)^T$ and $\mathbf{p}_{2,O} = (-1.2, 2.1, 0.8)^T$. The radii of the obstacles are given as $r_1 = 0.3$ m and $r_2 = 0.35$ m. Following the assumptions in Sec. II and Sec. III, we generate all possible module combinations from 2-DOF to 5-DOF: 3108 different compositions are generated. After generating all possible compositions, the tests in (B) are applied to all combinations and it is seen that only 21 of them are feasible regarding the initial and the goal positions.

In the next step, multiple shooting is applied to the remaining compositions from (B). Considering that the cost is defined as *the minimum execution time*, the objective function $L(\mathbf{x}_k(t), \mathbf{u}_k(t))$ is set as 1. Limits of each joint module $q_{min,max}$, $\dot{q}_{min,max}$ and $\tau_{min,max}$ are forwarded as lower and upper bounds to the optimization, which are also given in Tab. I. The initial and the final velocities of each joint are fixed to zero. As an initial estimate, the trajectories between the initial and the goal positions are considered as linear lines for each composition. We use 8 multiple shooting nodes which are determined by linear interpolation

of the initial and the final state values. Starting from the initial position, trajectories are generated from integrating the equation $\dot{\mathbf{x}}(t) = g(\mathbf{x}(t), \mathbf{u}(t))$ for each point until the goal position. The ODEs are solved for each of these points using *ode45* in MATLAB. The continuity condition explained in (C) is considered as a nonlinear equality constraint in the optimization. For the last ODE solution, the last point of the trajectory generated in the 7th node must be at the goal position, which is indicated in terminal constraints. To obtain the robot dynamics, the Newton-Euler equation in (1) is used. The obstacle avoidance constraints are defined as nonlinear inequality constraints and all collision possibilities in [11] are checked for each module.

The optimization process is done in MATLAB Optimization ToolboxTM using the *fmincon* solver for which we select the *interior-point* method. All compositions remaining from (B) are considered in the first two iterations and then we compare the results obtained from the iterations for all different compositions. The values of the state variables, time value, and control variables are normalized and the tolerances for them are set as 0.01. We compare the results in each two iterations and eliminate the compositions in case their cost values are 50% higher than the best result for the current simulation. Until reaching the optimal time values for all remaining compositions, this process is repeated in every iteration.

After 16 iterations, the optimal time is computed and only 4 different compositions remain at the end. Their cost values are compared to each other and it is seen that the optimal composition achieving the given task in minimum time is the composition $\kappa = B - J1 - L3 - J1 - L3 - J1 - L2 - J1 - L1 - E1$, which executes the given task within the minimum time as in Fig. 4.

We also implement the optimal trajectory generation method to all compositions achieving the tests in (B) individually. When we compare the total number of iterations for both methods, it is seen that the computational time for the proposed method is 35% more efficient when compared to the brute force method. It is also seen that the time-optimal composition is the same module composition for both methods. We also compare the optimized time values

TABLE I
THE MODULE PARAMETERS INVOLVED IN SIMULATION (SEE FIG.3)

	l	d	τ	q_{limits}	\dot{q}_{limits}	m	I	r
	[m]	[m]	[Nm]	[rad or m]	[rad/s or m/s]	[kg]	[kgm ²]	[m]
L1	0.75	0.2	-	-	-	0.05	-	-
L2	0.75	0.2	-	-	-	0.05	-	-
L3	0.75	0.2	-	-	-	0.05	-	-
J1	0.25	0.2	100	$[-\pi, \pi]$	$[-1, 1]$	1	$(2, 2, 0.65) \cdot 10^{-3}$	$(0, 0, 0)^T$
J2	0.25	0.2	90	$[0, 0.2]$	$[-5, 5]$	0.9	$(2, 2, 0.65) \cdot 10^{-3}$	$(0, 0, 0.1)^T$
EE1	0.2	0.2	95	$[-\pi, \pi]$	$[-1, 1]$	0.5	$(2, 2, 0.65) \cdot 10^{-3}$	$(0, 0, 0)^T$
EE2	0.2	0.2	85	$[0, 0.1]$	$[-5, 5]$	0.75	$(2, 2, 0.65) \cdot 10^{-3}$	$(0, 0, 0.025)^T$

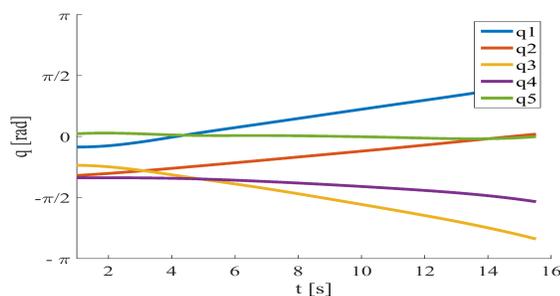


Fig. 4. Changes in the joint angles for the time-optimal composition

of the cost-optimal solution and the most cost-inefficient composition each composition and it is seen that the cost of the optimal composition is 45.2% of the most cost-inefficient composition.

V. CONCLUSION

In this paper, we present a task-based cost-optimal composition synthesis method for modular manipulators considering robot kinematics and dynamics. Unfeasible compositions are eliminated starting from easier tests to more difficult and more time-consuming ones. After implementing the kinematic tests, we apply direct multiple shooting to obtain the cost-optimal trajectories for the remaining compositions. Implementing the direct multiple shooting method to all remaining compositions would require significantly more time. To address this problem, we propose a method which is mainly based on the elimination of less-likely compositions considering the obtained cost values after each iteration. As a result, we decrease the total computational time while eliminating the compositions with high-cost values when compared to the other compositions. The results of our method are compared to results without implementing any elimination and it is seen that the proposed method is more time efficient. The main advantages of the proposed optimal composition synthesis algorithm are: *i*) it is applicable to different kinds of modules, *ii*) it avoids repeated compositions, *iii*) it considers the kinematics and dynamics, *iv*) it generates obstacle-free trajectories for each composition without violating the task requirements and the limits of the robots, and *v*) it provides a faster solution when compared to finding assemblies by optimizing trajectories for each assembly individually.

ACKNOWLEDGMENT

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme FP7/2007- 2013/ under REA grant agreement number 608022.

REFERENCES

[1] I.-M. Chen and J. W. Burdick, "Enumerating the non-isomorphic assembly configurations of modular robotic systems," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 702–719, 1998.

[2] Z. Bi, "On adaptive robot systems for manufacturing applications," Ph.D. dissertation, University of Saskatchewan, 2002.

[3] B. Dong and Y. Li, "Multi-objective-based configuration generation and optimization for reconfigurable modular robot," in *Proc. International Conference on Information Science and Technology (ICIST)*, 2011, pp. 1006–1010.

[4] S. Farritor, S. Dubowsky, N. Rutman, and J. Cole, "A systems-level modular design approach to field robotics," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 1996, pp. 2890–2895.

[5] W. K. Chung, J. Han, Y. Youm, and S. Kim, "Task based design of modular robot manipulator using efficient genetic algorithm," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 1, 1997, pp. 507–512.

[6] O. Chocron and P. Bidaud, "Evolutionary algorithms in kinematic design of robotic systems," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 1997, pp. 1111–1117.

[7] Q. Li and J. Zhao, "A universal approach for configuration synthesis of reconfigurable robots based on fault tolerant indices," *Industrial Robot: An International Journal*, vol. 39, no. 1, pp. 69–78, 2012.

[8] C. J. Paredis and P. Khosla, "Synthesis methodology for task based reconfiguration of modular manipulator systems," in *Proc. of the 6th International Symposium on Robotics Research, ISRR*, 1993.

[9] Z. Bi and W.-J. Zhang, "Concurrent optimal design of modular robotic configuration," *Journal of Robotic systems*, vol. 18, no. 2, pp. 77–87, 2001.

[10] J.-O. Kim and P. K. Khosla, "Design of space shuttle tile servicing robot: an application of task based kinematic design," in *Proc. IEEE International Conference on Robotics and Automation*, 1993, pp. 867–874.

[11] E. Icer, A. Giusti, and M. Althoff, "A task-driven algorithm for configuration synthesis of modular robots," in *Proc. IEEE International Conference on Robotics and Automation*, 2016, pp. 5203–5209.

[12] Z. Bi, W. A. Gruver, W.-J. Zhang, and S. Y. Lang, "Automated modeling of modular robotic configurations," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 1015–1025, 2006.

[13] I.-M. Chen and G. Yang, "Automatic model generation for modular reconfigurable robot dynamics," *Journal of dynamic systems, measurement, and control*, vol. 120, no. 3, pp. 346–352, 1998.

[14] A. Giusti and M. Althoff, "Automatic centralized controller design for modular and reconfigurable robot manipulators," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 3268–3275.

[15] Z. Shiller and S. Dubowsky, "Global time optimal motions of robotic manipulators in the presence of obstacles," in *Proc. IEEE International Conference on Robotics and Automation*, 1988, pp. 370–375.

[16] J. E. Bobrow, "Optimal robot plant planning using the minimum-time criterion," *IEEE Journal of Robotics and Automation*, vol. 4, no. 4, pp. 443–450, 1988.

[17] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast motions in biomechanics and robotics*. Springer, 2006, pp. 65–93.

[18] M. Gerdts, R. Henrion, D. Hömberg, and C. Landry, "Path planning and collision avoidance for robots," *Numerical Algebra, Control and Optimization*, vol. 2, no. 3, pp. 437–463, 2012.

[19] J. J. Craig, *Introduction to robotics: mechanics and control*. 2nd ed. Reading, MA: Addison-Wesley, 1989.

[20] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.