

Automatic Centralized Controller Design for Modular and Reconfigurable Robot Manipulators

Andrea Giusti and Matthias Althoff

Abstract—We address the problem of controlling modular robot manipulators. The challenge of modular-robot control is that the overall system dynamics are unknown due to its flexible composition from given modules. Most previous work has faced this problem by designing decentralized controllers. Simple decentralized controllers do not guarantee global asymptotic stability without knowledge of the overall system dynamics and alternative versions involving communication with neighboring modules result in complicated control concepts. Our approach is completely different: we store parameters regarding the dynamics and kinematics of each module and a unique identification number in itself. After finishing the assembly of the modules, the parameters are gathered in a central controller, which also detects the configuration using the identification numbers. Our centralized controller uses this information to synthesize model-based control laws on-the-fly as if the full system dynamics are known beforehand. We introduce a novel and compact notation to automate this procedure and to generalize the derivation of the kinematic and dynamic model for heterogeneous modules. Finally, a possible application is shown using simulations.

I. INTRODUCTION

Reconfigurable and modular robot manipulators are mechatronic systems composed of interchangeable modules. The modular nature of these systems allows them to be adapted for different applications, which is a clear advantage with respect to fixed-structure robots. This is especially useful in ultra-flexible environments, e.g. search and rescue operations, space explorations, service robots and robots for human-robot cooperation in manufacturing. In general, modular robots meet the different needs of the users through reconfigurations. Furthermore, modularity is also useful for robot manufacturers to reduce the number of parts as different robots can be assembled from a small set of modules. This makes it possible to provide a huge portfolio consisting of e.g. Selective Compliance Assembly Robotic Arm (SCARA) robots and anthropomorphic robots with a few standard modules.

In the last three decades several modular robot manipulators have been developed, for instance the RMMS [1], TOMMS [2], IRIS [3], PolyBot [4] and a spring assisted reconfigurable modular robot [5]. The high versatility of the modular and reconfigurable robot manipulators leads to several challenges, especially for the design of the control system. Considering arbitrary configurations of a nonuniform set of modules, an enormous number of different dynamic systems can be obtained, challenging the control design.

We present a new idea for controlling modular and reconfigurable robot manipulators based on distributed data stored in each module using a centralized control approach. After assembly of the modules, the modular information is collected by the central control unit, which automatically generates a centralized and model-based control law with guaranteed global asymptotic stability.

The control design for modular robot manipulators has been a longstanding problem in research. Most work has mainly focused on decentralized control approaches. Adaptive decentralized control methods for modular robots are presented in e.g. [6], [7]. In the former the dynamic model of each subsystem is approximated to cancel the couplings and in the latter the parameters of distributed proportional-integral-differential (PID) controllers are adjusted on-line. A decentralized control method for modular and reconfigurable manipulators is developed in [8] based on joint torque measurements for automatic compensation of the coupling effects. A decentralized and robust control method is presented in [9] that increases the performance with respect to a PID controller. A method for precision control of modular and reconfigurable manipulators with guaranteed stability is presented in [10] using the virtual decomposition approach. A hybrid architecture for centralized and decentralized operations is proposed in [11], where the authors consider the decentralized approach for controlling the manipulator dynamics.

Even though satisfying results have been achieved with advanced decentralized schemes, every decentralized control method can be considered as a special case of a centralized one. In principle, it is always possible to design a centralized controller with performance equal to or better than a decentralized control scheme. Especially when fast trajectory tracking is required or direct drive actuation is employed, centralized control schemes are superior compared to decentralized controllers since subsystem couplings can be compensated instead of treating them as disturbance [12]. Moreover, centralized architectures are also beneficial for optimal control, compliance and impedance control [12], dynamic scaling of trajectories [13], and failure detection [14].

Automatic generation of complete models from modules has previously been investigated in [15]. In [16] a centralized control design is considered. In contrast to our work, these approaches assume similar modules with symmetric geometry and a central database storing the information of all modules, which is impractical as discussed later. A previous approach that considered the storage of information in the

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

modules is described in [17], where the automatic derivation of kinematics is considered for joint modules with similar geometry.

Our proposed approach is different because we consider modules with arbitrary kinematic and dynamic parameters stored in each module. Our kinematic modeling procedure is based on the Denavit-Hartenberg convention, which is related to the works in [18]–[20]. In these works, it is required to store homogeneous transformation matrices in a central database, which is not a requirement of our approach. We introduce a novel notation and an extension of the standard Denavit-Hartenberg convention that simplifies the automatic procedure for obtaining the relative parameters, especially when considering prismatic joints. Moreover, our approach also considers dynamics with the final goal of automatically designing model-based control laws. The main novelties that this work introduces are summarized as follows: *i*) none of the cited approaches have considered distributing information to the modules to enable and/or automate the centralized and model-based controller design after information collection; *ii*) our automatic controller design approach is generalized and can thus be applied when modules are heterogeneous (e.g. with different shapes and dynamic parameters); *iii*) we provide a systematic method to characterize modules using a novel notation and an extension of the standard Denavit-Hartenberg convention; *iv*) our approach does not require symbolic computation for the automatic centralized controller design of reconfigurable modular robot manipulators.

The paper is organized as follows. In Sec. II we describe the control problem for modular robot manipulators in detail. The proposed method is presented in Sec. III followed by simulation results in Sec. IV.

II. PROBLEM DESCRIPTION

We consider a modular and reconfigurable robot manipulator composed of serially connected rigid links. Throughout this paper we also assume that:

- each module includes a rigid proximal part, a rigid joint, and a rigid distal part (see Fig. 3);
- the connectors are standardized and allow the assembly of two subsequent modules at only one relative orientation;
- the motor inertia effects are not considered for a succinct presentation of the material since this does not affect the basic idea of our approach.

For a concise presentation of our method, we further exclude modules that do not introduce a new degree of freedom (e.g. link modules). The consideration of such modules is trivial, as it becomes evident later.

A reconfigurable modular robot manipulator with N serially connected modules constitutes an open kinematic chain and has the following dynamic model (see [12]):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{f}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{u}, \quad (1)$$

where $\mathbf{q} \in \mathbb{R}^N$ is the vector of the generalized coordinates, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{N \times N}$ is the symmetric and positive definite mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ (with $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{N \times N}$) is the vector of the Coriolis and centrifugal terms, $\mathbf{f}(\dot{\mathbf{q}}) \in \mathbb{R}^N$ and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^N$

are respectively the vectors of friction and gravity terms, finally $\mathbf{u} \in \mathbb{R}^N$ is the vector of the actuation forces/torques. We consider only viscous friction without loss of generality of our presented approach. The matrices and vectors of the dynamic model in (1) are different for various compositions of the manipulator and type of modules.

We face the problem of automatically generating a centralized controller in the joint space that guarantees global asymptotic stability after a reconfiguration:

$$\lim_{t \rightarrow \infty} \|\mathbf{e}(t)\| = 0,$$

where $\|\mathbf{e}(t)\| = \|\mathbf{q}_r(t) - \mathbf{q}(t)\|$ is the Euclidean norm of the error vector in the joint space and $\mathbf{q}_r(t)$ is the differentiable (of class C^2) desired trajectory. We consider the development in time of this error as a performance indicator.

III. PROPOSED METHOD

Our proposed approach is illustrated in Fig. 1: each module is first characterized according to our proposed notation, which is stored within the module. After the selection of the modules and the manual assembly of the robot, the automatic controller design process starts. First, information related to the kinematics, dynamics and a unique identification number (e.g. the Media Access Control (MAC) address) are collected from all modules by the central control unit. Next, the centralized controller is automatically generated to let the robot operate with guaranteed motion control performance.

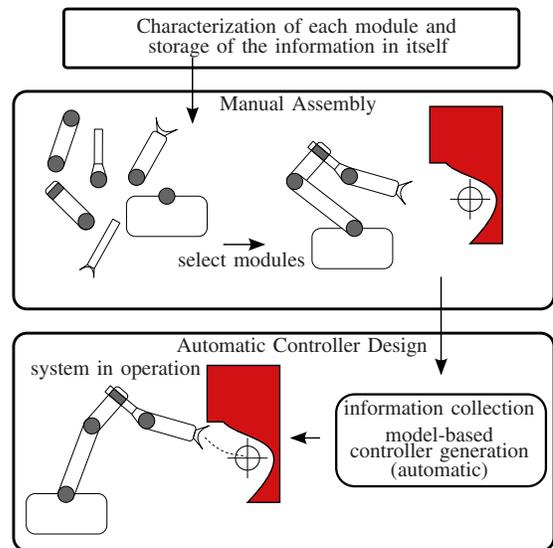


Fig. 1. Illustration of the proposed approach.

The information of the modules can be stored in two ways: *i*) using a centralized database with stored information of all the modules that can be used; *ii*) using distributed memory to store information of a module in the module itself. Obviously, the latter option is preferred since the first one requires database updates when new modules are created and additionally requires storage of all possible modules (leading to storage and legacy issues).

We propose a tree structure as a suitable topology of the communication network for modular robotic manipulators with an open kinematic chain. The tree-structured network supports serial and branch-structured manipulators and is composed of a coordinator associated with the central control unit, routers associated with the intermediate modules and end-devices associated with the end effectors. In this structure, each module allows the coordinator to get measurements (e.g. joint position and velocity), set input commands to the actuators and set/get data in/from its local memory database (e.g. dynamic and kinematic parameters and type of joints). Moreover, collecting additional information (e.g. the routing tables) makes it possible for the central control unit to detect the robot configuration.

Appropriate information from the modules enables the central control unit to automatically compute the forward kinematics based on an extended Denavit and Hartenberg convention as is described in III-A and the dynamic model with the recursive Newton-Euler method as presented in III-B. Finally, model-based control laws are automatically synthesized as discussed in III-C.

A. Forward Kinematics using Modular Information

The forward kinematics allows obtaining the pose of the end effector (position and orientation) given the joint positions. For an open kinematic chain the forward kinematics is commonly derived in a recursive way, by multiplying the homogeneous transformation matrices that relate the frame of reference of each link to the previous one. A systematic method for the assignment of the frame of reference of each link is the Denavit-Hartenberg (D-H) convention [21], which we briefly recall.

Consider that the i^{th} link of the robot has a body-fixed frame of reference with axes x_i , y_i and z_i . According to the standard D-H convention (see Fig. 2), z_i is aligned with the axis of the joint at the connection with link $i + 1$. The x_i axis is chosen along the common normal between the axis z_{i-1} and z_i pointing toward the link $i + 1$. The origin of the frame i is set at the intersection of the common normal with z_i . Finally, the y_i axis completes the right-handed coordinate system. Once the frames of reference are assigned, four parameters are introduced to define the relative transformation of coordinates: a_i , d_i , α_i , θ_i . As shown in Fig. 2, a_i is the distance along x_i between the origin of the frame i' and i , d_i is the distance along z_{i-1} between the origin of the frame $i - 1$ and i' , α_i is the angle between the axis z_{i-1} and z_i around x_i and θ_i the angle between x_{i-1} and x_i around z_{i-1} (the angles are positive counterclockwise). The only variable parameter is d_i when the joint is prismatic and θ_i when it is revolute [12].

Considering that our proposed approach is automatic and that the standard D-H convention is not unique, we extend it to handle some particular cases that need special consideration (see Fig. 2):

- when the z axes intersect, the x_i unit vector is obtained from the cross product between them;

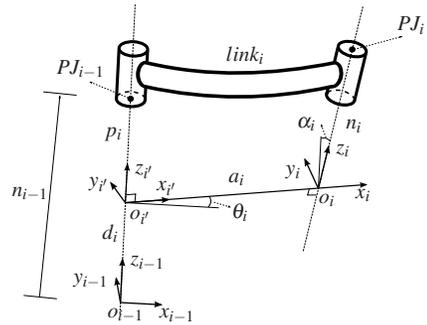


Fig. 2. Representation of a link showing the parameters for kinematics.

- when the z axes are parallel the x_i unit vector is set along the common normal between them and the origin o_i is set at the joint connection PJ_i ;
- when the z axes are superimposed the x_i unit vector is aligned with x_{i-1} and the origin o_i is set at the joint connection PJ_i .

In order to automate the derivation of the four D-H parameters for each link (D-H table), we introduce two additional parameters associated to each link. These are obtained on the basis of the link geometry: p_i and n_i , and are the z coordinates of the point PJ_{i-1} from $o_{i'}$ and of the point PJ_i from o_i , respectively (see Fig. 2). Using these two parameters, d_i can be recursively computed as:

$$d_i = n_{i-1} - p_i \text{ (revolute joint),}$$

$$d_i = n_{i-1} - p_i + q_i \text{ (prismatic joint),}$$

where q_i is the joint displacement. Additionally, in order to capture the constant angular offset that can be present between x_{i-1} and x_i when the joint is in its zero position and when it is prismatic, an additional parameter γ_i is introduced. With this addition, the D-H parameter θ_i is:

$$\theta_i = \gamma_i + q_i \text{ (revolute joint),}$$

$$\theta_i = \gamma_i \text{ (prismatic joint).}$$

In the following, we introduce a notation to characterize arbitrary modules and the automatic procedure to obtain the parameters of the extended D-H convention (a_i , α_i , p_i , n_i , γ_i) for each link of an assembled manipulator. Let us consider the exemplary module shown in Fig. 3(a). This module is composed of an input connector, a joint and an output connector. To define the new notation, we first have to decompose the module into a proximal (pl) and a distal part (dl). These parts are shown in Fig. 3(c) and Fig. 3(b), respectively. In this description, we consider modules with only one degree of freedom without loss of generality because more complex modules can be modeled as a series of those considered, by setting the relevant parameters to zero.

The first step to characterize the module is to fix a frame of reference for the proximal part and the distal part. These two frames are located in the center of the interfaces of the respective connectors. The axes x_{in} and x_{out} are positioned

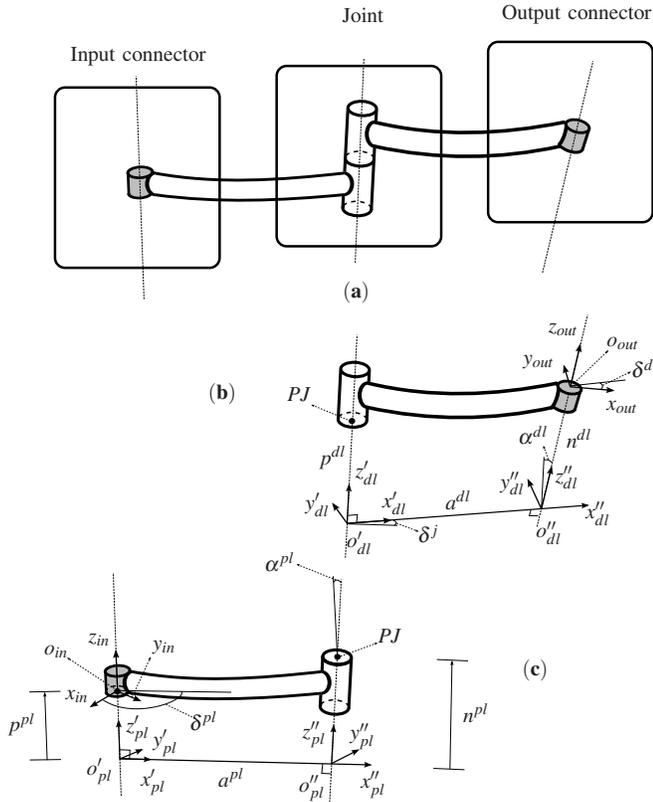


Fig. 3. Kinematic notation for module characterization. The connectors are indicated in light-grey color, (a) is the entire module, (b) the distal part and (c) the proximal part.

along a unique and standardized direction on the connection plane. The z axes are normal to the respective connector planes, z_{in} points in towards the input connector and z_{out} points outwards from the output one (see Fig. 3). The axes y_{in} and y_{out} are selected to complete the respective right-handed frames of reference. According to our notation we characterize both the proximal and the distal part with a set of parameters. To obtain these parameters, the same approach of the extended D-H convention described previously is applied. Accordingly, we obtain four parameters for the proximal part: a^{pl} , α^{pl} , p^{pl} , n^{pl} , and for the distal part: a^{dl} , α^{dl} , p^{dl} and n^{dl} .

Regarding the proximal part illustrated in Fig. 3(c), two auxiliary frames are considered. The first one has its origin at o'_{pl} , the intersection of the common normal between z_{in} and the joint axis with z_{in} . The axis x'_{pl} is set along the common normal pointing toward the distal part, z'_{pl} is set along z_{in} and y'_{pl} completes the right-handed frame of reference. The second auxiliary frame has its origin in o''_{pl} at the intersection of the common normal with the joint axis, its axis x''_{pl} is set along the common normal and points toward the distal part, z''_{pl} is set along the joint axis and finally y''_{pl} completes the right-handed frame of reference. The four parameters for the proximal part have the following meanings:

- a^{pl} is the distance between o'_{pl} and o''_{pl} along the common normal;

- α^{pl} is the angle between the axis z_{in} and the joint axis around x''_{pl} ;
- p^{pl} is the z coordinate of the input connection point o_{in} from o'_{pl} ;
- n^{pl} is the z coordinate of the joint connection point PJ from o''_{pl} .

For the distal part illustrated in Fig. 3(b), two auxiliary frames are analogously introduced. The four parameters for the distal part have similar meanings:

- a^{dl} is the distance between o'_{dl} and o''_{dl} , along the common normal;
- α^{dl} is the angle between the joint axis and z_{out} around x''_{dl} ;
- p^{dl} is the z coordinate of the joint connection point PJ from o'_{dl} ;
- n^{dl} is the z coordinate of the output connection point o_{out} from o''_{dl} .

Finally, three additional parameters are required: δ^{pl} , δ^{dl} , δ^j . As shown in Fig. 3(c) and (b): δ^{pl} is the angle between x'_{pl} and x_{in} , δ^{dl} is the angle between x''_{dl} and x_{out} , and finally δ^j is the angle between x''_{pl} and x'_{dl} when the joint is in its zero position (all the angles of this notation are positive counterclockwise). Particular cases of the relative orientation between the z axes (e.g. parallel, intersect or overlap) are handled similarly to the previously described extension of the standard D-H convention. The connection points to consider are: o_{in} and PJ (proximal part); PJ and o_{out} (distal part). We collect the parameters required to characterize a module for kinematics in Tab.I.

TABLE I
INFORMATION STORED IN EACH MODULE FOR KINEMATICS.

Proximal	a^{pl}	α^{pl}	p^{pl}	n^{pl}	δ^{pl}
Distal	a^{dl}	α^{dl}	p^{dl}	n^{dl}	δ^{dl}
Joint	δ^j		Joint type		

It is worth noting that the same approach can be applied to modules with multiple input and output connectors. In those cases, a set of parameters for each realizable combination of the connectors is required and has to be stored in the module. When the information is forwarded to the central control unit, the set of parameters corresponding to the connectors in use are sent.

Using the proposed notation, the derivation of the parameters of the extended D-H convention for each link (a_i , α_i , p_i , n_i , γ_i) can be automated. Let us assume that the i^{th} connection between a module $i-1$ and a module i is established. In order to obtain the parameters for the link, we require the homogeneous transformation matrix \mathbf{F}_i of a frame oriented as the D-H one located at PJ_i (see Fig. 2) with respect to a frame parallel to the first auxiliary frame of the distal part of the module $i-1$ with origin PJ_{i-1} (see Fig. 3(b)). From now on, we use a compact notation to indicate operations with homogenous transformation matrices. For instance: $T_k(\cdot)/R_k(\cdot)$ are homogeneous transformations that represent

the translation/rotation along/around the k axis. Using the parameters that characterize subsequent modules, we first calculate an auxiliary matrix \mathbf{F}'_i . This matrix describes the transformation of a frame with origin at PJ_i , parallel to the second auxiliary frame of the proximal part of module i , with respect to a frame parallel to the first auxiliary frame of the distal part of module $i-1$ and with origin at PJ_{i-1} . It is calculated as:

$$\begin{aligned} \mathbf{F}'_i &= T_z(-p_{i-1}^{dl})T_x(a_{i-1}^{dl})R_x(\alpha_{i-1}^{dl})T_z(n_{i-1}^{dl}) \\ &\quad R_z(\delta_{i-1}^{dl} - \delta_i^{pl})T_x(-p_i^{pl})T_x(a_i^{pl})R_x(\alpha_i^{pl})T_z(n_i^{pl}) \\ &= \begin{bmatrix} \mathbf{R}'_i & \mathbf{U}'_i \\ \mathbf{0}^T & 1 \end{bmatrix}. \end{aligned} \quad (2)$$

In order to complete the synthesis matrix \mathbf{F}_i , an additional possible rotation ϕ_i around z_i is considered. In fact, we need to align the x axis according to our extended D-H convention. With the transformation of (2) we only reached a frame located at PJ_i with the orientation of the second auxiliary frame of the proximal part of module i . An additional rotation around z may be required to bring this frame parallel to the D-H one. In order to find this angle, we have to consider three possible cases for the joint axes. The detection of the case that occurs is performed using \mathbf{U}'_i , and the unit vector of the z axis in \mathbf{R}'_i , which contains the information of the relative orientation of two subsequent joint axes.

- i. When the axes overlap: $\phi_i = 0$.
- ii. When the axes are parallel: $\phi_i = \arctan(v_y/v_x)$, from $\mathbf{V} = [v_x \ v_y \ v_z]^T = \mathbf{R}'_i{}^T \mathbf{U}'_i$.
- iii. When the axes are skew or intersect: $\phi_i = \arctan(v_y/v_x)$, from $\mathbf{V} = [v_x \ v_y \ v_z]^T = \mathbf{R}'_i{}^T \mathbf{V}'$ and $\mathbf{V}' = \mathbf{z}_i \times \mathbf{z}_{i-1}$.

The synthesis matrix is completed as:

$$\mathbf{F}_i = \mathbf{F}'_i R_z(\phi_i) = \begin{bmatrix} r_{x_i} & s_{x_i} & t_{x_i} & u_{x_i} \\ r_{y_i} & s_{y_i} & t_{y_i} & u_{y_i} \\ r_{z_i} & s_{z_i} & t_{z_i} & u_{z_i} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

The same transformation of \mathbf{F}_i can be expressed in terms of the parameters of a link and an auxiliary rotation (ϕ') as:

$$\begin{aligned} \mathbf{L}_i &= T_z(-p_i)R_z(\phi'_i)T_x(a_i)R_x(\alpha_i)T_z(n_i) \\ &= \begin{bmatrix} C_{\phi'_i} & -C_{\alpha_i}S_{\phi'_i} & S_{\alpha_i}S_{\phi'_i} & a_iC_{\phi'_i} + n_iS_{\alpha_i}S_{\phi'_i} \\ S_{\phi'_i} & C_{\alpha_i}C_{\phi'_i} & -S_{\alpha_i}C_{\phi'_i} & a_iS_{\phi'_i} - n_iS_{\alpha_i}C_{\phi'_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & n_iC_{\alpha_i} - p_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (4)$$

where $S_\varepsilon/C_\varepsilon$ are the abbreviations of sine/cosine of an angle ε . Equating (4) and (3), the parameters of our extended D-H convention for the i^{th} connection (or link) can be inferred:

$$\begin{aligned} a_i &= u_{x_i} r_{x_i} + u_{y_i} r_{y_i}, \\ \alpha_i &= \arctan(s_{z_i}/t_{z_i}), \\ \phi'_i &= \overbrace{\arctan(r_{y_i}/r_{x_i})}^{\phi'_i} + \delta_{i-1}^j - \phi_{i-1}. \end{aligned}$$

If $s_{z_i} \neq 0$:

$$\begin{aligned} n_i &= \frac{(u_{x_i} r_{y_i} - u_{y_i} r_{x_i})}{s_{z_i}}, \\ p_i &= \frac{(u_{x_i} t_{z_i} r_{y_i} - u_{z_i} s_{z_i} - u_{y_i} t_{z_i} r_{x_i})}{s_{z_i}}. \end{aligned}$$

If $s_{z_i} = 0$:

$$\begin{aligned} n_i &= 0, \\ p_i &= -u_{z_i}. \end{aligned}$$

B. Dynamic Model from Modular Information

The problem of deriving the dynamic model of a rigid link manipulator is usually solved either with Lagrangian formulation [22] or recursive Newton-Euler (N-E) methods [23]. A comparison of the methods for the derivation of the equations of motion of a rigid link manipulator is in [22]. A computationally efficient variant of the Newton-Euler method has first been proposed in [23] with a complexity that grows linearly with the number of the joints ($O(N)$). Subsequently, enhanced versions of this algorithm have been presented in e.g. [24], [25]. A review on methods for robot dynamics can be found in [26].

The recursive N-E algorithm studies the manipulator link by link and is composed of two recursions, one for kinematics (forward recursion) and one for forces and torques (backward recursion). It can be used either with symbolic computations to obtain the closed form dynamic model or with numerical ones for control as described later. We consider the recursive N-E algorithm of [12] without motor inertia effects. Similar to the kinematics, we provide a systematic approach to characterize the modules for dynamics. The set of required parameters for each module is presented as well as the procedure to synthesize them for each established connection. For each link of the manipulator (see Fig. 4), a body-fixed frame of reference with origin in \mathbf{D}_i is considered. We consider this frame to have the same orientation as the D-H one so that the results of the forward kinematics are exploited to infer relative orientation of subsequent links. With reference to the generic link of Fig. 4, the vector from \mathbf{D}_{i-1} to \mathbf{D}_i is $\mathbf{r}_{D_{i-1}, D_i}^i$ and the vector from \mathbf{D}_i to \mathbf{C}_i (the center of mass) is \mathbf{r}_{D_i, C_i}^i . In addition to these vectors, the mass m_i , the inertia tensor \mathbf{I}_i^i and the viscous damping coefficient of the joint FJ_{i-1} are also required for the algorithm. The superscript of all the vectors indicates in which frame of reference they are expressed.

Similarly to the kinematics, a synthesis procedure is required for the dynamics based on modular information. A graphical representation of the i^{th} connection involving dynamic parameters is shown in Fig. 5. In our approach, the required parameters for dynamics are expressed in the output frame for the distal part and in the input frame for the proximal part. For each module, beyond the mass for both the distal (m^{dl}) and the proximal part (m^{pl}), we require their inertia tensors: \mathbf{I}_{dl}^{out} , \mathbf{I}_{pl}^{in} , and the coordinates of the centers of mass: \mathbf{r}_{Cdl}^{out} , \mathbf{r}_{Cpl}^{in} , expressed in the output and the input frame, respectively.

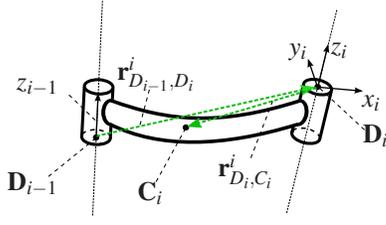


Fig. 4. Representation of a link for dynamic related parameters.

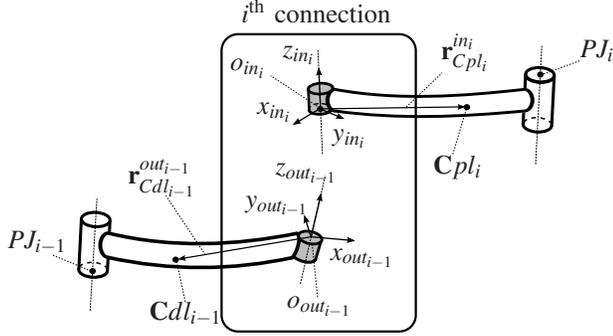


Fig. 5. Representation of a connection involving parameters for dynamics. Connectors are indicated in light-grey color.

Using this notation, when the i^{th} connection is realized, the output frame of module $i-1$ and the input frame of module i match. We denote the matched frame as “io”. The synthesis of the total mass of the link m_i , the coordinates of its center of mass $\mathbf{r}_{C_i}^{io}$ and the inertia tensor \mathbf{I}_i^{io} can thus be automated:

$$m_i = m_{i-1}^{dl} + m_i^{pl}, \quad \mathbf{I}_i^{io} = \mathbf{I}_{dl_{i-1}}^{out} + \mathbf{I}_{pl_i}^{in},$$

$$\mathbf{r}_{C_i}^{io} = \frac{m_{i-1}^{dl} \mathbf{r}_{C_{dl_{i-1}}}^{out} + m_i^{pl} \mathbf{r}_{C_{pl_i}}^{in}}{m_i}.$$

Our proposed automatic procedure transforms these quantities to be expressed in a frame parallel to the D-H one and located at \mathbf{D}_i for each realized link i (see Fig. 4). Since the kinematic parameters are known, these transformations of coordinates are performed using homogeneous transformations and Steiner’s theorem [12] for the inertia tensor. The coordinates of the center of mass are computed as:

$$\begin{bmatrix} \mathbf{r}_{D_i, C_i}^i \\ 1 \end{bmatrix} = [\mathbf{A}_i^{io}]^{-1} \begin{bmatrix} \mathbf{r}_{C_i}^{io} \\ 1 \end{bmatrix},$$

where

$$\mathbf{A}_i^{io} = R_z(-\delta_i^{pl})T_z(-p_i^{pl})T_x(a_i^{pl})R_x(\alpha_i^{pl})T_z(n_i^{pl})R_z(\phi_i)$$

$$= \begin{bmatrix} \mathbf{R}_i^{io} & \mathbf{U}_i^{io} \\ \mathbf{0}^T & 1 \end{bmatrix}.$$

The inertia tensor is computed as:

$$\mathbf{I}_i^i = \mathbf{R}_i^{ioT} \left(\mathbf{I}_i^{io} - m_i \mathbf{S}^T(\mathbf{r}_{C_i}^{io}) \mathbf{S}(\mathbf{r}_{C_i}^{io}) \right) \mathbf{R}_i^{io} + m_i \mathbf{S}^T(\mathbf{r}_{D_i, C_i}^i) \mathbf{S}(\mathbf{r}_{D_i, C_i}^i),$$

where $\mathbf{S}(\cdot)$ is an anti-symmetric matrix:

$$\mathbf{S}(\mathbf{U}) = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}, \quad \text{with } \mathbf{U} = [u_x \ u_y \ u_z]^T.$$

The last parameter that needs to be synthesized for each constituted link is $\mathbf{r}_{D_{i-1}, D_i}^i$. Also for this vector we can use homogeneous transformations with the synthesized kinematic parameters:

$$\mathbf{A}_i^{i-1} = T_z(-p_i)R_z(\gamma_i + q_i)T_x(a_i)R_x(\alpha_i)T_z(n_i) \quad (\text{rev. joint}),$$

$$\mathbf{A}_i^{i-1} = T_z(-p_i + q_i)R_z(\gamma_i)T_x(a_i)R_x(\alpha_i)T_z(n_i) \quad (\text{prism. joint}),$$

and finally

$$\mathbf{A}_{i-1}^i = [\mathbf{A}_i^{i-1}]^{-1} = \begin{bmatrix} \mathbf{R}_{i-1}^i & \mathbf{U}_{i-1}^i \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad \mathbf{r}_{D_{i-1}, D_i}^i = -\mathbf{U}_{i-1}^i.$$

We store the additional parameters in each module for the dynamics as collected in Tab. II.

TABLE II
INFORMATION TO STORE IN THE MODULES FOR DYNAMICS.

Proximal	m^{pl}	\mathbf{I}_{pl}^{in}	$\mathbf{r}_{C_{pl}}^{in}$
Distal	m^{dl}	\mathbf{I}_{dl}^{out}	$\mathbf{r}_{C_{dl}}^{out}$
Joint	FJ		

C. Centralized Controller Design

The procedures described for automatic derivation of kinematics and dynamics enable the automatic generation of model-based control laws using the modular information in Tab. I and II. Our proposed centralized controller for modular and reconfigurable robot manipulators takes as input the position vector (\mathbf{q}), the velocity vector ($\dot{\mathbf{q}}$), the reference trajectory (\mathbf{q}_r , $\dot{\mathbf{q}}_r$, $\ddot{\mathbf{q}}_r$) and finally the array of structures containing the information of the modules: **ModRob**. The array of structures **ModRob** contains the information of Tab. I and II for each module. The output is the closed loop force/torque command, which guarantees global asymptotic stability. In order to realize the model-based control laws, we have implemented a function $\mathbf{u} = NE_{mod}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{ModRob})$, that performs the recursive N-E algorithm using **ModRob** as described in Sec. III-A and III-B and returns the force/torque commands.

We have implemented the automatic generation of two model-based control laws: proportional and derivative (PD) control with gravity compensation and computed torque control. Computed torque control is one of the most effective centralized and model-based control methods for robotic manipulators with rigid links. Assuming perfect knowledge of the system dynamics, computed torque linearizes the model of (1) by compensating the nonlinear and coupling terms through feedback. The control law is completed by assigning an input (v) to the linearized system that guarantees asymptotically stable dynamics of the error in joint space [27]. The PD control with gravity compensation [12] is suitable for point to point motion and is designed to compensate the gravitational terms of the dynamic model and to include

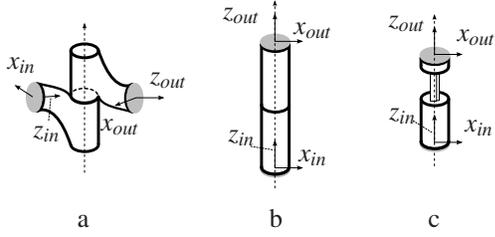


Fig. 6. Representation of the modules for illustrative purpose used for simulations. Connectors are indicated in light-grey color.

TABLE III
INFORMATION OF THE MODULES USED FOR SIMULATION.

Parameter	a	b	c
a^{pl} (m)	0	0	0
α^{pl} (rad)	$-\pi/2$	0	0
p^{pl} (m)	-0.1	-0.1	-0.1
n^{pl} (m)	0	0	0
δ^{pl} (rad)	0	0	0
a^{dl} (m)	0	0	0
α^{dl} (rad)	$-\pi/2$	0	0
p^{dl} (m)	0	-0.1	-0.05
n^{dl} (m)	0.1	0	0
δ^{dl} (rad)	0	0	0
δ^j (rad)	π	0	0
Joint Type	Rev.	Rev.	Prism.
$\mathbf{r}_{C_{pl}}^m$ (m)	$[0 \ 0 \ 0.05]^T$	$[0 \ 0 \ 0.05]^T$	$[0 \ 0 \ 0.05]^T$
m^{pl} (kg)	0.5	0.5	0.5
\mathbf{I}_{pl}^m (kgm^2)	10^{-3} .	10^{-3} .	10^{-3} .
(diagonal)	$[2 \ 2 \ 0.65]$	$[2 \ 2 \ 0.65]$	$[2 \ 2 \ 0.65]$
$\mathbf{r}_{C_{dl}}^{out}$ (m)	$[0 \ 0 \ -0.05]^T$	$[0 \ 0 \ -0.05]^T$	$[0 \ 0 \ -0.025]^T$
m^{dl} (kg)	0.5	0.5	0.25
\mathbf{I}_{dl}^{out} (kgm^2)	10^{-3} .	10^{-3} .	10^{-3} .
(diagonal)	$[2 \ 2 \ 0.65]$	$[2 \ 2 \ 0.65]$	$[0.36 \ 0.36 \ 0.31]$
FJ (Nms)	3	3	3

a linear feedback control law for positions and velocities. Assuming perfect knowledge of the gravitational term, PD control with gravity compensation guarantees global and asymptotic stability for any desired constant equilibrium posture [12].

We obtain the compensating commands for the control laws numerically by varying appropriate numerical arguments to the function NE_{mod} . For example, the numerical vector of the gravitational terms used for PD control with gravity compensation is computed as $\mathbf{g}(\mathbf{q}) = NE_{mod}(\mathbf{q}, \mathbf{0}, \mathbf{0}, \text{ModRob})$ and the closed loop command of the computed torque control is computed numerically as $\mathbf{u}_{cl} = NE_{mod}(\mathbf{q}, \dot{\mathbf{q}}, v, \text{ModRob})$ where $\mathbf{q}, \dot{\mathbf{q}}$ are the current numerical vectors of positions and velocities.

IV. SIMULATION RESULTS

A point to point motion in the joint space is simulated for a configuration using a given set of modules. We show and compare results of the automatically generated control laws: PD control with gravity compensation and computed torque control. Gaussian measurement noise and input disturbance are included. We consider these distributions with zero mean and standard deviation of $\sigma_n = 0.03$ for measurements of joint positions (*rad* and *m*) and velocities (*rad/s* and *m/s*)

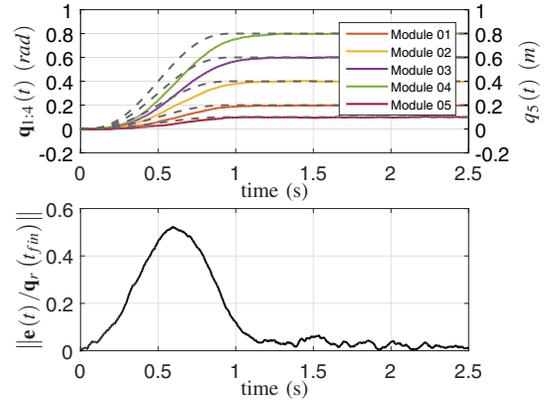


Fig. 7. Simulation using PD with gravity compensation controller. The dashed lines represent the desired trajectories.

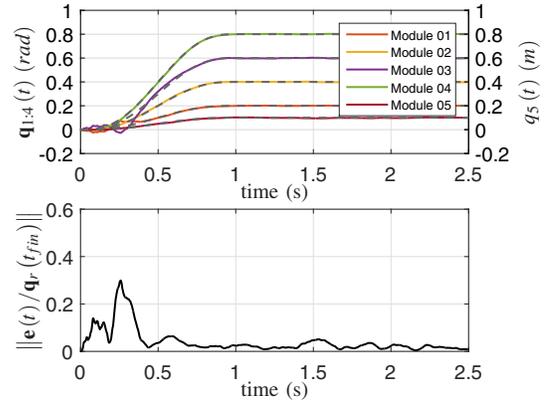


Fig. 8. Simulation using computed torque controller. The dashed lines represent the desired trajectories.

and $\sigma_u = 0.3$ for the input forces/torques (*N* and *Nm*). We perform the simulations using three kind of modules shown in Fig. 6: a, b (revolute joints) and c (prismatic joint). The parameters of the modules used for these examples are collected in Tab. III. Using these modules we build a configuration composed of five modules: [b-a-b-a-c]. The trajectory of each joint is designed with a quintic polynomial. The coefficients have been selected to guarantee zero initial/final velocity and zero initial/final acceleration. The final time of the planned motion is $t_{fin} = 1s$ with a starting time $t_{in} = 0s$. The joints start from zero to reach the final positions: $\mathbf{q}(t_{fin}) = [0.2, 0.4, 0.6, 0.8, 0.1]^T$. The simulations are performed using the following gain matrices [12] of the controllers: $\mathbf{K}_p = 250\mathbf{I}$ and $\mathbf{K}_d = 30\mathbf{I}$, where \mathbf{I} is the identity matrix of proper dimensions.

As shown in Fig. 7 and Fig. 8, computed torque control has better performance observing the development in time of the Euclidean norm of the error. In these figures, we show the error vector as a fraction of the respective final positions of the trajectories for normalized presentation in one plot.

V. CONCLUSIONS

We present a systematic and effective approach for centralized control of modular and reconfigurable robot manip-

ulators. To the best knowledge of the authors, none of the approaches presented in the literature so far have considered distributing information to the modules to automatically generate a pure model-based centralized controller, especially when considering heterogeneous modules. This idea paves the way for future modular and reconfigurable robotic systems with performance and capabilities that approach those of fixed structure ones from a control perspective. Having the possibility to auto-generate a model-based centralized controller with guaranteed asymptotic stability for a non-uniform set of modules, the central control unit does not require a large database and successive updates. Fast reconfiguration with guaranteed stable operations and motion control performance ensures hyper-flexibility of the robotic system.

A drawback of this approach compared to conventional methods for controlling modular robot manipulators is the necessary storage of information in the modules that may increase manufacturing costs. Additionally, if self-detection of the configuration were required, network solutions that support a tree-structured topology would be needed. A communication bus (e.g. CAN bus) could also be suitable but additional communication lines have to be employed as described in [17]. However, we believe that the cost savings from standardization of modular robots would outweigh such cost increases. Furthermore, the automated generation of the controller with guaranteed motion control performance would reduce the time and costs required for commissioning the robot after assembling or reconfiguration.

An important continuation of this work is the introduction of robustness for uncertain modular information, elasticity in the joints and measurement noise, especially because in future works this control approach will be implemented on a real robot.

ACKNOWLEDGMENT

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

REFERENCES

- [1] D. Schmitz, P. Khosla, and T. Kanade, "The CMU reconfigurable modular manipulator system," Inst. Software Res., Carnegie Mellon Univ., Pittsburgh, PA, USA, CMU-RI-TR-88-7, Tech. Rep., 1988.
- [2] T. Matsumaru, "Design and control of the modular robot system: TOMMS," in *Proc. 1995 IEEE Int. Conf. Robotics and Automation*, vol. 2, May 1995, pp. 2125–2131.
- [3] R. Hui, N. Kircanski, A. Goldenberg, C. Zhou, P. Kuzan, J. Wiercinski, D. Gershon, and P. Sinha, "Design of the IRIS facility—a modular, reconfigurable and expandable robot test bed," in *Proc. 1993 IEEE Int. Conf. Robotics and Automation*, vol. 3, May 1993, pp. 155–160.
- [4] M. Yim, D. Duff, and K. Roufas, "PolyBot: a modular reconfigurable robot," in *Proc. 2000 IEEE Int. Conf. Robotics and Automation*, vol. 1, 2000, pp. 514–520.
- [5] G. Liu, Y. Liu, and A. Goldenberg, "Design, analysis, and control of a spring-assisted modular and reconfigurable robot," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 4, pp. 695–706, Aug 2011.
- [6] M. Zhu and Y. Li, "Decentralized adaptive fuzzy control for reconfigurable manipulators," in *2008 IEEE Conference Robotics, Automation and Mechatronics*, Sept 2008, pp. 404–409.
- [7] W. Melek and A. Goldenberg, "Neurofuzzy control of modular and reconfigurable robots," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 3, pp. 381–389, Sept 2003.
- [8] G. Liu, S. Abdul, and A. Goldenberg, "Distributed modular and reconfigurable robot control with torque sensing," in *Proc. 2006 IEEE Int. Conf. Mechatronics and Automation*, June 2006, pp. 384–389.
- [9] W. Li, Z. Melek and C. Clark, "Decentralized robust control of robot manipulators with harmonic drive transmission and application to modular and reconfigurable serial arms," *Robotics*, vol. 27, pp. 291–302, 2009.
- [10] W.-H. Zhu, T. Lamarche, E. Dupuis, D. Jameux, P. Barnard, and G. Liu, "Precision control of modular robot manipulators: The VDC approach with embedded FPGA," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1162–1179, Oct 2013.
- [11] D. Wang, A. Goldenberg, and G. Liu, "Development of control system architecture for modular and re-configurable robot manipulators," in *ICMA 2007. Int. Conf. Mechatronics and Automation*, Aug 2007, pp. 20–25.
- [12] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2009.
- [13] J. Hollerbach, "Dynamic scaling of manipulator trajectories," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 106, pp. 102–106, 1984.
- [14] A. De Luca and R. Mattone, "Actuator failure detection and isolation using generalized momenta," in *Proc. 2003 IEEE Int. Conf. Robotics and Automation*, vol. 1, Sept 2003, pp. 634–639.
- [15] I. Chen, S. Yeo, G. Chen, and G. Yang, "Kernel for modular robot applications: Automatic modeling techniques," *The International Journal of Robotic Research*, vol. 18, no. 2, pp. 225–242, 1999.
- [16] E. Meister, A. Gutenkunst, and P. Levi, "Dynamics and control of modular and self-reconfigurable robotic systems," *International Journal on Advances in Intelligent Systems*, vol. 6, no. 1&2, pp. 66–78, 2013.
- [17] P. Xinan, W. Hongguang, J. Yong, and X. Jizhong, "Automatic kinematic modelling of a modular reconfigurable robot," *Transactions of the Institute of Measurement and Control*, vol. 35, no. 7, pp. 922–932, 2013.
- [18] L. Kelmar and P. Khosla, "Automatic generation of kinematics for a reconfigurable modular manipulator system," in *Proc. 1988 IEEE Int. Conf. Robotics and Automation*, vol. 2, Apr 1988, pp. 663–668.
- [19] B. Benhabib, G. Zak, and M. Lipton, "A generalized kinematic modeling method for modular robots," *Journal of Robotic Systems*, vol. 6, no. 5, pp. 545–571, 1989.
- [20] Z. Bi, W. Zhang, I. Chen, and S. Lang, "Automated generation of the D-H parameters for configuration design of modular manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 23, pp. 553–562, 2007.
- [21] J. Denavit and R. Hartenberg, "A kinematic notation of lower-pair mechanisms based on matrices," *ASME Journal of Applied Mechanics*, vol. 22, pp. 215–221, 1955.
- [22] J. Hollerbach, "A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 10, no. 11, pp. 730–736, Nov 1980.
- [23] J. Luh, M. Walker, and R. Paul, "On-line computational scheme for mechanical manipulators," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 102, pp. 468–474, 1980.
- [24] A. De Luca and L. Ferrajoli, "A modified Newton-Euler method for dynamic computations in robot fault detection and control," in *Proc. 2009 IEEE Int. Conf. Robotics and Automation*, May 2009, pp. 3359–3364.
- [25] L. Sciacivco, B. Siciliano, and L. Villani, "Lagrange and Newton-Euler dynamic modelling of a gear-driven rigid robot manipulator with inclusion of motor inertia effects," *Advanced Robotics*, vol. 10, no. 3, pp. 317–334, 1996.
- [26] R. Featherstone and D. Orin, "Robot dynamics: equations and algorithms," in *Proc. 2000 IEEE Int. Conf. Robotics and Automation*, vol. 1, 2000, pp. 826–834.
- [27] W. Chung, L.-C. Fu, and S.-H. Hsu, "Motion control," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 133–159.