JOURNAL OF THEORETICAL
AND APPLIED MECHANICS
62, 3, pp. 615-629, Warsaw 2024
https://doi.org/10.15632/jtam-pl/190056

MOTION PLANNING FOR TASK-BASED MOTIONS OF MECHANICAL SYSTEMS BASED ON COMPUTATIONALLY GENERATED REFERENCE DYNAMICS

Elżbieta Jarzębowska
Warsaw University of Technology, Warsaw, Poland
corresponding author Elżbieta Jarzębowska, e-mail: elajarz@meil.pw.edu.pl
Krzysztof Augustynek, Andrzej Urbaś
University of Bielsko-Biała, Bielsko-Biała, Poland
e-mail: kaugustynek@ubb.edu.pl; aurbas@ubb.edu.pl

The paper presents a development of a complementary motion planning strategy for task-based motions for mechanicl systems. The key component of the strategy is a computational procedure for generation of constrained dynamical models, where constraints can be material or task-based ones and specify work regime requirements. The procedure provides the constrained dynamics, i.e. reference dynamics, whose solutions satisfy all constraints upon systems and enable motion planning. It is a unified tool for constrained motion analysis, motion planning and controller designs. The procedure effectiveness is demonstrated through constrained dynamics generation and motion planning analysis for a robotic system.

Keywords: task based motion planning, task based constraints, constrained multibody dynamics

1. Introduction

1.1. The paper scope and content

The paper presents a development of a complementary motion planning strategy for mechanical systems, e.g. robotic ones, which are dedicated to deliver work and services, and these can be pre-planned by task based constraints formulated by constraint equations. The key component of the presented motion planning strategy is a computational based procedure for generation of constrained system dynamical models (Jarzębowska *et al.*, 2018a). The constraints imposed upon system models may be holonomic and first order nonholonomic, material or nonmaterial, and the latter ones are referred to as programmed. The programmed constraints, which reflect variety of performance requirements put upon mechanical systems, combined with other constraints imposed upon them, are merged into one constrained dynamics, referred to as a reference dynamics, whose solutions satisfy all constraints put upon them. Then, motion subjected to the desired constraints may be analyzed, refined and planned accordingly. Due to holonomic and nonholonomic constraints of different natures, which are imposed upon various mechanical systems, the paper proposes a novel approach to the motion planning for constrained systems, which is dedicated to task based motions. Also, the approach based upon the reference dynamics aims to support designing of tracking or stabilizing controllers for programmed motions.

1.2. Motion planning strategies developed for mechanical systems

Motion planning is one of the most significant activities for many mechanical systems like robotic systems, either stationary or mobile, car-like vehicles, heavy machine equipment like cranes and many others. In majority of works, motion planning refers to the planning of motion between the start and final locations and the satisfy constraints. The constraints are mostly collision-avoidance ones. Much of this research and satisfy the constraints has been focused on solving the motion planning problem in a stationary environment where both targets and obstacles are stationary; see e.g. (La Valle, 2006) and references there. A more complex case is for mobile robots and car-like vehicles, where nonholonomic constraints have to be taken into account in the planner as well as overall dynamics. There are many approaches to determine feasible paths for such systems, see e.g. the review in (Lu *et al.*, 2016). Different approaches for motion planning for car-like vehicles and mobile robots operating in dynamic environments are proposed. According to (Chiang *et al.*, 2015), they can be classified as: artificial potential fields-based approaches, state-time space based approaches, velocity obstacles-based motion planning approaches, and probabilistic collision checking based approaches. The approaches which are based on the artificial potential field generate a combined potential field in which the vehicle is attracted to its target position and is kept away from the obstacles The combined potential field moves the vehicle in the work space prespecifying both position and full stop after obstacles velocity (Chiang *et al.*, 2015; Bounini *et al.*, 2017). In this approach, a vehicle is often modelled as a particle and obstacles are convex shaped. Some of the drawbacks of this approach, listed the most often, are possibility of falling into local minima and intensive computation which is required for descriptions of real environment for vehicles. The state-time space approach is an extension of the configuration space and it consists of notation of the vehicle position and time (Chiang *et al.*, 2015). In this approach, the vehicle is modelled as a particle and stationary, and moving obstacles are transformed to static ones. The approaches which are based on the velocity obstacles concept define the velocity obstacles by computing vehicle velocities that would cause a collision with the obstacles assuming that they are moving with constant velocities (Lee *et al.*, 2017). Then, an avoidance maneuver is computed by selecting velocities that are outside of the velocity obstacles and a derived collision-free trajectory consists of a series of avoidance maneuvers (Huang *et al.*, 2018). However, these approaches are restricted to planning decisions to the velocity of the vehicle and they represent each obstacle, including the vehicle, as a disk. These approaches also require strict determination of the time horizon; otherwise the vehicle may skip moving through tight spaces, and then it can be difficult to determine the shortest path. The selection of the proper time horizon is still unresolved. In (van den Berg and Overmars, 2008), probabilistic approaches are used for computing paths in the state-time space. These planners incrementally build a tree of explored configurations for each planning query. Latest research results demonstrate attempts for reducing the complexity of the planning problem by first constructing a path based on the off-line information of the environment, and then planning a collision-free trajectory on this path/roadmap that considers the on-line information (Feyzabadi and Caprin, 2016). There are also approaches which consider uncertainties or imperfections in environment representation performing probabilistic collision detection. Other problems related to finding a path for the vehicle which is safe by construction can be found in, e.g. (Feyzabadi and Caprin, 2016) and references there.

1.3. Methods for dynamics generation for mechanical systems

In our approach to task-based motion planning, the central role plays the constrained dynamics developed by a computational procedure, in which all constraints i.e. holonomic and nonholonomic, material or task-based are merged together. This constrained dynamics, referred to as reference dynamics, can be developed for both stationary and mobile car-like vehicles or robots. Obstacles, fixed or moving, can be treated as constraints and their locations should be specified by the constraints. There are two basic distinctions between the presented computational procedure for reference dynamics development and other constrained motion equations derivation methods, usually based upon the Lagrange approach and its modifications. The first one is that all constrants upon a system can be considered and merged into one dynamical model and the second should be: that all constraints upon is that the final equations are in the reduced state form, i.e. they are free of constraint reaction forces, which are eliminated at the derivation process. These are essential advantages of our approach, and this computational procedure serves both reference and control oriented dynamics derivation. It works for rigid and flexible system models, for open and closed-loop kinematic chains and enables automation of constrained dynamics generation (Jarzębowska *et al.*, 2018b, 2023). The computational procedure for the reference dynamics development is the one unified method for including any desired trajectory or velocity functions into system dynamics. This distinguishes our method from others developed to obtain dynamics of constrained systems, which are based mostly on Newton-Euler and Lagrange approaches. In many approaches, trajectories are planned separately of the vehicle model by using constraints for position, velocity and acceleration at each time instant like in (Macfarlane and Croft, 2003). Some dynamics modeling approaches exploit the Udwadia-Kalaba derivation method, see, e.g. (Yang *et al.*, 2019; Liu and Liu, 2016) and references there. The so called Udwadia-Kalaba dynamic equations enable including position and first order constraints into the system dynamics and determining the constraint reaction forces. These forces, which ensure that the constraints are satisfied, are referred to as control forces. However, from the control theory point of view, it may be not very convenient to design a control algorithm realizing these forces because they needed to be determined via analytical equations directly from the Udwadia-Kalaba equations and may depend upon higher orders of e.g. velocities. Also, Kane's equations can be used to derive dynamic models of robotic systems, see e.g. (Kane and Lewinson, 1983; Sayers, 1990). They eliminate the nonworking constraint reaction forces in an elegant way but they require a smart choice of generalized speeds, which rely on a modeler experience and may not be straightforward as inputs for control applications. In the case when one wants to select velocities as motion parameters, the use of Boltzmann-Hamel equations written in quasi-velocities can be a good option for dynamics and the use of the Boltzmann-Hamel equations control algorithm generations, see e.g. (Jarzębowska and Cichowski, 2018). Due to complexity of mechanical systems, including robotic systems, intensive computation dedicated to their dynamics derivation, solutions and motion analysis is required. For example, in (Khalil *et al.*, 2017), a recursive approach, based upon Newton-Euler equations, of tree-structure systems with rigid and flexible links with floating bases based upon the Newton-Euler is proposed. It can be applied numerically or using symbolic techniques. An algorithm to generate the inverse dynamics is presented in (Do *et al.*, 2021), and it is based upon the recursive Newton-Euler algorithm, the chain rule of differentiation and the computer algebra.

1.4. The paper motivation and contribution

Our goals are to develop a computationally based procedure for generation constrained dynamical models that would support planning of programmed motions and designing tracking controllers for executing these motions specified by the programmed constraints. These goals are motivated by usually used methods of motion planning and controller designs for constrained dynamical models reported in literature, see e.g. (Dixon *et al.*, 2003) and references there. Usually, system dynamical models without constraints are generated first and next control goals like tracking predefined trajectories or other desired motions, are specified. Next, a controller is designed and quite often, it is dedicated to this dynamics and an associated control goal.

In our approach, the reference dynamics describes the system behavior when the task-based constraints are on, and it may serve two purposes. The first one is analysis and assessment of kinematic parameters needed to be reached by the system to follow the desired motions. It is enabled by solutions of the reference dynamics. If this motion comes from work or service requirements put upon an existing system, it can be easily verified whether it is feasible and realistic to be accomplished by this system. The second purpose is to provide the complementary motion planner, in which the outputs of the reference dynamics are inputs to the controller. The tracking or stabilizing control architecture can be designed and it uses motions planned by the reference dynamics.

The paper is organized as follows. After the introductory Section, Section 2 provides the details of derivation of the computational procedure of generating constrained system dynamics, which provides motion planning for system models. In Section 3, it is demonstrated how the planned motion can be implemented to a control platform architecture for tracking these desired motions. Section 4 details an example of desired motion of a mechanical system model, e.g. a three-link manipulator model subjected to task-based constraints. Simulation studies about the task-based motion planning are detailed in Section 5. We analyse the planned task-based motion and provide an example of a controller design. Other controllers can be designed in the same way. The paper closes with conclusions, future research prospects and the list of references.

2. The computational procedure of generating constrained system dynamics for mechanical system models

The constraints put on a system are referred to as programmed and they can be combined together with material ones. They are imposed as control goals on system performance or as service tasks, and they all can be presented in a general form (Jarzębowska, 2012, 2008)

$$\mathbf{B}(t, \mathbf{q}, \dot{\mathbf{q}}, \dots, \mathbf{q}^{(p-1)})\mathbf{q}^{(p)} + \mathbf{s}(t, \mathbf{q}, \dot{\mathbf{q}}, \dots, \mathbf{q}^{(p-1)}) = \mathbf{0} \tag{2.1}$$

The constraints can be material for $p = 0, 1$, or nonmaterial, i.e. programmed for $p \geqslant 1$. The nonmaterial constraints are imposed by a designer or a control engineer to obtain a system desired performance, e.g. they can be imposed upon acceleration $p = 2$ or jerk $p = 3$, as well as for desired trajectories with $p = 1$. Constraints form (2.1) is the generalized constraint formulation and it encompasses the classical analytical constraint concept.

The generalized programmed motion equations (GPME) for rigid body models subjected to constraints (2.1) have the following form (Jarzebowska, 2008)

$$\begin{aligned} &\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{Q}(t, \mathbf{q}, \dot{\mathbf{q}}) \\ &\mathbf{B}(t, \mathbf{q}, \dot{\mathbf{q}}, \dots, \mathbf{q}^{(p-1)})\mathbf{q}^{(p)} + \mathbf{s}(t, \mathbf{q}, \dot{\mathbf{q}}, \dots, \mathbf{q}^{(p-1)}) = \mathbf{0} \end{aligned} \tag{2.2}$$

where: $\mathbf{M}(\mathbf{q})$ is the mass matrix, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ is a vector of centrifugal forces, $\mathbf{g}(\mathbf{q})$ is a vector of gravity forces, $\mathbf{Q}(t, \mathbf{q}, \dot{\mathbf{q}})$ is a vector of external generalized forces which are not controls.

Constrained dynamics model (2.2) is the reference dynamics. The solutions to the reference dynamics satisfy constraints (2.1) imposed upon the model. Equations (2.2) are free of constraint reaction forces, which are eliminated in the derivation process. This is the fundamental advantage of (2.2) which makes them suitable for direct motion analysis, planning task-based motions and for control applications. More details about derivation and application of the GPME method are available in (Jarzębowska, 2012). The derivation of GPME (2.2) requires determination of the system kinetic energy and its derivatives. The derivation algorithm for them is as follows: Assume that constraint equations in (2.2) may be solved, at least locally, with respect to the vector $\mathbf{q}_\beta^{(p)}$ of dependent coordinates, i.e.

$$\mathbf{q}_\beta^{(p)} = \mathbf{g}_\beta^{(p)}(t, q, \dot{q}, \dots, q_\mu^{(p)}) \tag{2.3}$$

and $q = (q_\beta, q_\mu)$, $q_\beta \in \mathbb{R}^k$, $q_\mu \in \mathbb{R}^{n-k}$. Then do the following:

1. Construct a function P_p such that $P_p = (1/p)[T^{(p)} - (p+1)T_0^{(p)}]$, where T is kinetic energy of an unconstrained system, $T^{(p)}$ is its p-th order time derivative, and $T_0^{(p)}$ is defined by $T_0^{(p)} = \sum_{\sigma=1}^{n}(\partial T)/(\partial q_\sigma)q_\sigma^{(p)}$.
2. Construct a function R_p such that $R_p = P_p - \sum_{\sigma=1}^{n} Q_\sigma q_\sigma^{(p)} = R_p(t, q, \dot{q}, \dots, q_\mu^{(p)}, q_\beta^{(p)}, q^{(p+1)})$.
3. Construct a function R_p^*, in which $q_\beta^{(p)}$ from R_p are replaced by constraints form (2.3)

$$R_p^* = R_p^*(t, q, \dot{q}, \dots, q_\mu^{(p)}, g_\beta^{(p)}(t, q, \dot{q}, \dots, q_\mu^{(p)}), q^{(p+1)}) = R_p^*(t, q, \dot{q}, \dots, q_\mu^{(p)}, q^{(p+1)})$$

4. Assuming that components of a vector of external forces satisfy $\partial Q_\sigma/\partial q_\sigma^{(p)} = 0$, equations of the generalized programmed motion equations (GPME) for a system with constraints (2.3) have the form

$$\frac{\partial R_p^*}{\partial q_\mu^{(p)}}\Big|_{\mu=k+1,\dots,n} = \frac{\partial R_p}{\partial q_\mu^{(p)}} + \sum_{\beta=1}^{k} \frac{\partial R_p}{\partial q_\beta^{(p)}} \frac{\partial g_\beta^{(p)}}{\partial q_\mu^{(p)}} = 0 \tag{2.4}$$

   Resulting equations (2.4) plus constraints (2.3) are equivalent to (2.2).

This derivation procedure can be nicely automated in Matlab or other software environment tools, however, when friction, compliance, flexibility or other phenomena are included into the model, this derivation algorithm may be time consuming. For these reasons, for engineering applications, a computationally efficient approach to the constrained dynamics derivation is needed. To develop the automated and computer oriented procedure for generation of constrained dynamics for rigid body models, reference dynamics (2.2) is rewritten in the form (Jarzębowska *et al.*, 2018)

$$\begin{aligned} &\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q}) + \mathbf{Q}(t, \mathbf{q}, \dot{\mathbf{q}}) \\ &\mathbf{B}(t, \mathbf{q})\dot{\mathbf{q}} = -\mathbf{s}(t, \mathbf{q}) \end{aligned} \tag{2.5}$$

where the matrices and vectors are designed as

$$\mathbf{M}(\mathbf{q}) = \mathbf{M}_i\Big|_{i\in i_c} + \sum_{j\in i_{d_c}} \mathbf{M}_j \frac{\partial \dot{q}_j}{\partial \dot{q}_i} \qquad \mathbf{M} = \sum_{l=1}^{n_l} \mathbf{M}^{(l)} \qquad \mathbf{M}^{(l)} = (m_{ij}^{(l)})_{i,j=1,\dots,n_{dof}}$$

$$m_{ij}^{(l)} = \operatorname{tr}(\mathbf{T}_i^{(l)}\mathbf{H}^{(l)}(\mathbf{T}_j^{(l)})^{\mathrm{T}}) \qquad \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{h}_i\Big|_{i\in i_c} + \sum_{j\in i_{d_c}} \mathbf{h}_j \frac{\partial \dot{q}_j}{\partial \dot{q}_i} \qquad \mathbf{h} = \sum_{l=1}^{n_l} \mathbf{h}^{(l)}$$

$$h_i^{(l)} = \sum_{m=1}^{n_{dof}^{(l)}} \sum_{n=1}^{n_{dof}^{(l)}} \operatorname{tr}(\mathbf{T}_m^{(l)}\mathbf{H}^{(l)}(\mathbf{T}_{m,n}^{(l)})^{\mathrm{T}})\dot{q}_m^{(l)}\dot{q}_n^{(l)} + 2\sum_{m=1}^{n_{dof}^{(l)}} \sum_{n=1}^{n_{dof}^{(l)}} \operatorname{tr}(\mathbf{T}_m^{(l)}\mathbf{H}^{(l)}(\mathbf{T}_{i,n}^{(l)})^{\mathrm{T}})\dot{q}_m^{(l)}\dot{q}_n^{(l)}$$

$$\mathbf{h}^{(l)} = (h_i^{(l)})_{i=1,\dots,n_{dof}^{(l)}} \qquad \mathbf{g}(\mathbf{q}) = \mathbf{g}_i\Big|_{i\in i_c} + \sum_{j\in i_{d_c}} \mathbf{g}_j \frac{\partial \dot{q}_j}{\partial \dot{q}_i}$$

$$\mathbf{g} = \sum_{l=1}^{n_l} \mathbf{g}^{(l)} \qquad \mathbf{g}^{(l)} = (g_i^{(l)})_{i=1,\dots,n_{dof}^{(l)}} \qquad g_i^{(l)} = m^{(l)} g \mathbf{J}_1 \mathbf{T}_i^{(l)} \mathbf{r}_{C^{(l)}}^{(l)}$$

$$\mathbf{Q}(t, \mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}_i\Big|_{i\in i_c} + \sum_{k\in i_c\cup i_{d_c}} \dot{q}_k \frac{\partial Q_k}{\partial \dot{q}_i} + \sum_{j\in i_{d_c}} \Big(\mathbf{Q}_j + \sum_{k\in i_c\cup i_{d_c}} \dot{q}_k \frac{\partial Q_k}{\partial \dot{q}_j}\Big) \frac{\partial \dot{q}_j}{\partial \dot{q}_i}$$

$$\mathbf{Q} = \sum_{l=1}^{n_l} \mathbf{Q}^{(l)} \qquad \mathbf{Q}^{(l)} = (Q_i^{(l)})_{i=1,\dots,n_{dof}^{(l)}}$$

and $\mathbf{T}^{(l)}$ is a transformation matrix from the local frame of link l to the global reference frame $\{0\}$

$$\mathbf{T}_i^{(l)} = \frac{\partial \mathbf{T}^{(l)}}{\partial q_i^{(l)}} \qquad \mathbf{T}_{i,j}^{(l)} = \frac{\partial^2 \mathbf{T}^{(l)}}{\partial q_i^{(l)} \partial q_j^{(l)}}$$

$\mathbf{H}^{(l)}$ is a pseudo-inertia matrix, $m^{(l)}$ is mass of the link, $\mathbf{r}_{C^{(l)}}^{(l)}$ is the position vector of the center of mass.

In the procedure, it is assumed that constraint Eq. $(2.5)_2$ for $p = 0, 1$ may be solved, at least locally, with respect to the vector $\dot{\mathbf{q}}_{d_c}$ of dependent coordinates

$$\dot{\mathbf{q}}_{d_c} = \dot{\mathbf{q}}_{d_c}(t, \mathbf{q}, \dot{\mathbf{q}}_{i_c}) \tag{2.6}$$

and $\dot{\mathbf{q}} = [\dot{\mathbf{q}}_{i_c}^{\mathrm{T}}, \dot{\mathbf{q}}_{d_c}^{\mathrm{T}}]^{\mathrm{T}}$, $\dot{\mathbf{q}}_{d_c} \in \mathbb{R}^{n_{d_c}}$, $\dot{\mathbf{q}}_{i_c} \in \mathbb{R}^{n_{dof} - n_{d_c}}$.

This partition to independent and dependent coordinate derivatives is equivalent to selection of control inputs at the stage of controller design. Comparing (2.5) derivation to the original GPME derivation reported herein and resulted in (2.4), it can be seen that the computational based procedure for generation of constrained system dynamical models developed in this paper is computer oriented ready to automation. In our derivation, which differs from many schemes based upon the Newton-Euler or Lagrange approaches, we have applied the formalism of joint coordinates and homogeneous transformation matrices together with the matrix trace concept. They enable the effective automatic generation of matrices and vectors for the modification of the GPME algorithm. Also, the selection of independent coordinates which will serve as control inputs conforms to the proactive approach to dynamics and control design for mechanical system models (Banaszuk *et al.*, 2007).

3. Planning and tracking task based motions – an advanced control platform architecture

The reference dynamics, either in general form (2.2) or specialized (2.5) offers advantages from both constrained dynamics motion analyses, planning, and controller design points of view. The constrained dynamics, i.e. the reference dynamics, when solved, shows motion patterns of the planned motion. When they reflect work regime or other engineering related demands, it is easy to analyze kinematic characteristics of a system under the constraint action and other accompanying phenomena like vibration. It enables concluding whether the constraints imposed by a designer or control engineer are realistic for the analyzed system and can be accomplished. They can also be modified accordingly. With the aid of our reference dynamics, i.e. the motion planner for constraint driven motion, a controller can be designed based upon a tracking strategy architecture, presented in Fig. 1. It is dedicated for tracking constrained motions. The key component of the tracking strategy is the computational based procedure for generation constrained system dynamics and constrained motion planning (reference dynamics – motion planning block). Based upon experience with the GPME approach to tracking, various modeling parameters can be selected and the dynamic control model can be developed using any available mechanics based method (dynamic control model block) (Jarzębowska, 2008, 2009). Control laws can be selected from collections of algorithms from linear or nonlinear control theory. It is reflected by the block of "specialized terms to the control law". For example, non--adaptive or adaptive controllers can be used. Both dynamic models, i.e. the reference and control, can be derived in a coordinate system convenient for a designer (Jarzębowska, 2009).

To demonstrate the ease of applying motion planning and then tracking with the strategy architecture presented in Fig. 1, let us adopt feedback linearization to the control dynamics.

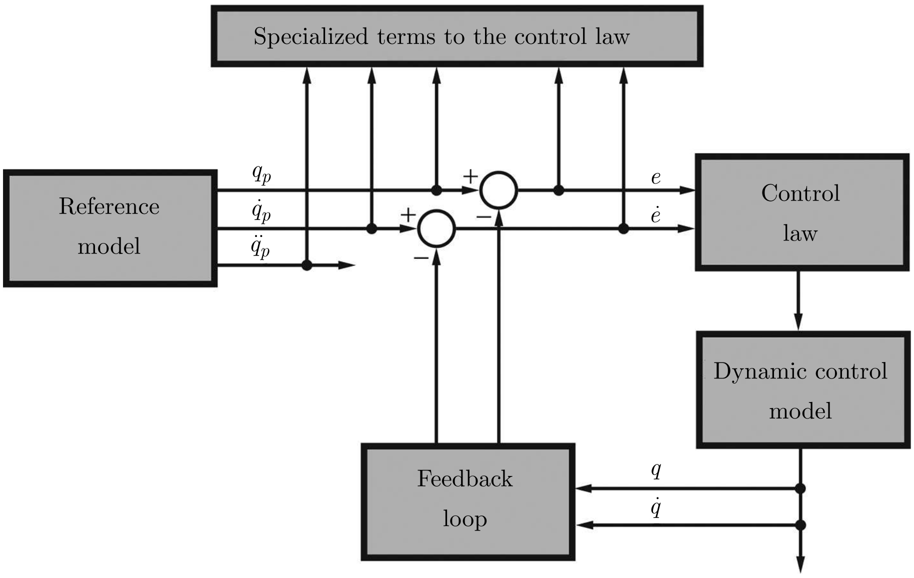


Fig. 1. Tracking strategy architecture

First, generate the reference dynamics (motion planning block), then consider the dynamics with no constraints on the system model (dynamic control model block), i.e.

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \tag{3.1}$$

Then, apply feedback linearization which enables replacing the controller vector $\boldsymbol{\tau}$ by a virtual controller vector $\mathbf{u}$ as

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\mathbf{u} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \tag{3.2}$$

where the vector $\mathbf{u} = (u_i)_{i=1,\dots,n_{dof}}$. For the linearized dynamic control model

$$\ddot{\mathbf{q}} = \mathbf{u} \tag{3.3}$$

the controller $\mathbf{u}$ can be selected. For a purpose of illustration, let us pick the PD controller of the form

$$u_i = \ddot{\widehat{q}}_i - 2\delta_i \dot{e}_i - \delta_i^2 e_i \tag{3.4}$$

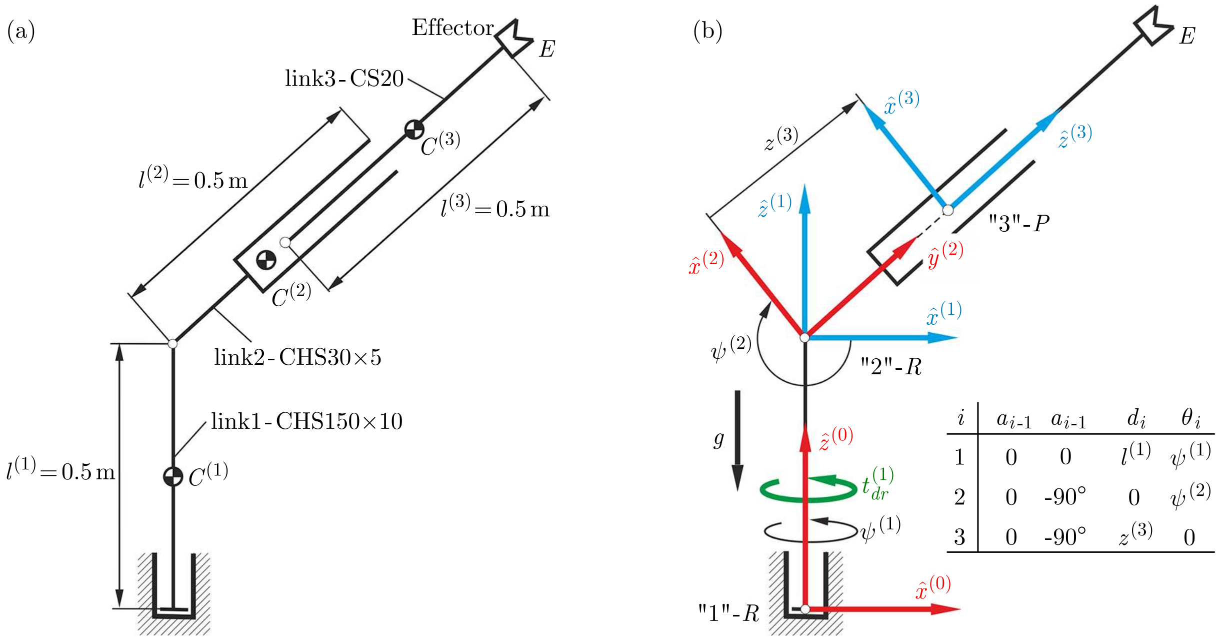
where $e_i = q_i - \widehat{q}_i$ is the tracking error, $\widehat{q}_i$ is a value of the i-th coordinate obtained form the motion planner, and δ_i is the control gain.

Notice that within this control architecture, control dynamics (3.1) can be transformed to another control form and another controller can be designed.

4. Example – three-link manipulator task – based motion planning

To demonstrate the process of generation of the reference dynamics, programmed motion planning, its analyses and the controller design, let us examine an example of a three link manipulator whose physical model is presented in Fig. 2a. In Fig. 2b, Denavit-Hartenberg parameters to the manipulator motion description are selected. Motion of the manipulator is described by the vector of generalized coordinates as

$$\mathbf{q} = (q_j)_{j=1,\dots,3} = [\psi^{(1)}, \psi^{(2)}, z^{(3)}]^{\mathrm{T}} \tag{4.1}$$



i	$a_{i\text{-}1}$	$a_{i\text{-}1}$	d_i	θ_i
1	0	0	$l^{(1)}$	$\psi^{(1)}$
2	0	-90°	0	$\psi^{(2)}$
3	0	-90°	$z^{(3)}$	0

Fig. 2. (a) Model of a three-link manipulator. (b) The Denavit-Hartenberg notation

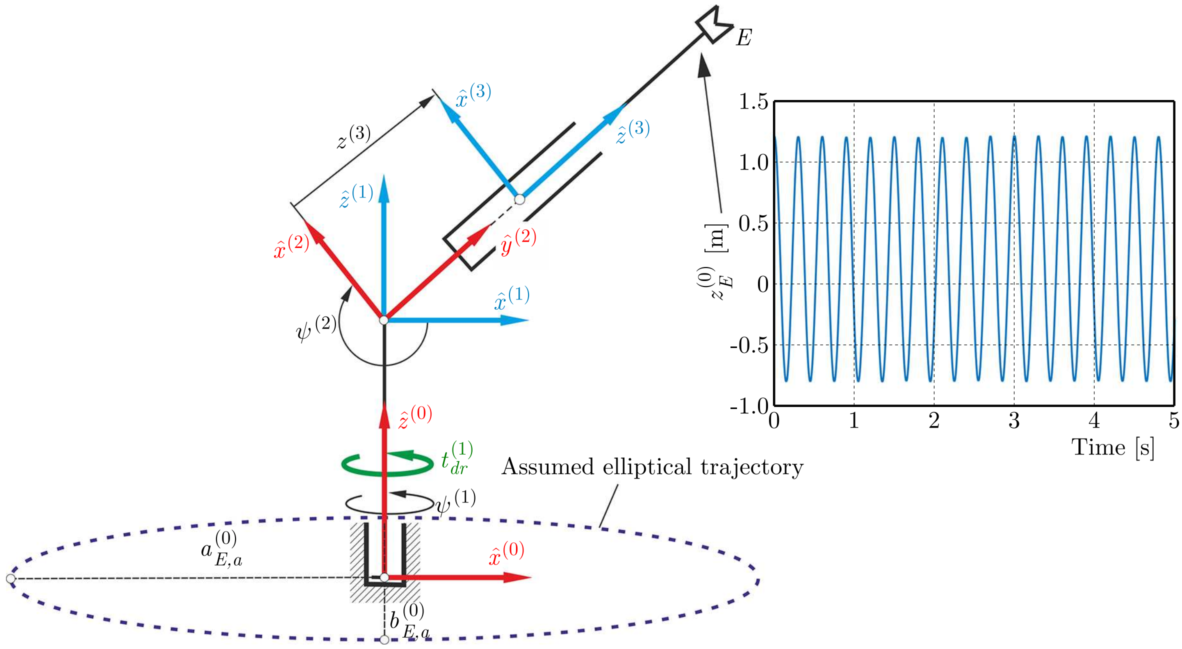


Fig. 3. Assumed programmed constraints

Programmed constraint equations are imposed due to desired motion of the manipulator end-effector. It is illustrated in Fig. 3. The programmed constraints formulation is as follows

$$
\begin{aligned}
&\varPhi_1 \equiv 0 \;\Rightarrow\; \Big(\frac{x_E^{(0)}}{a_{E,a}^{(0)}}\Big)^2 + \Big(\frac{y_E^{(0)}}{b_{E,a}^{(0)}}\Big)^2 - 1 = 0 \\
&\varPhi_2 \equiv 0 \;\Rightarrow\; z_E^{(0)} - z_{E,a}^{(0)} = 0
\end{aligned}
\tag{4.2}
$$

where: $x_E^{(0)} = \mathbf{J}_1\mathbf{T}^{(3)}\mathbf{r}_E^{(3)}$, $y_E^{(0)} = \mathbf{J}_2\mathbf{T}^{(3)}\mathbf{r}_E^{(3)}$, $z_E^{(0)} = \mathbf{J}_3\mathbf{T}^{(3)}\mathbf{r}_E^{(3)}$, $a_{E,a}^{(0)}$, $b_{E,a}^{(0)}$ are half-axes of the programmed elliptical trajectory, $z_{E,a}^{(0)}$ is the time assumed function.

It can be seen that this specific kind of the end-effector motion may reflect work regime for the manipulator. Other constraints can be formulated in the same way. The derivation procedure requires differentiation of the programmed constraint equations. The derivatives take the form

$$\begin{aligned}&\dot{\varPhi}_1 \equiv 0 \;\Rightarrow\; \mathbf{u}\dot{\mathbf{q}} = \mathbf{0} \qquad &&\dot{\varPhi}_2 \equiv 0 \;\Rightarrow\; \mathbf{C}_3\dot{\mathbf{q}} = 0\\ &\ddot{\varPhi}_1 \equiv 0 \;\Rightarrow\; \mathbf{u}\ddot{\mathbf{q}} + v = 0 \qquad &&\ddot{\varPhi}_2 \equiv 0 \;\Rightarrow\; \mathbf{C}_3\ddot{\mathbf{q}} + d_3 = 0\end{aligned} \tag{4.3}$$

where

$$\begin{aligned}\mathbf{C} &= \begin{bmatrix}\mathbf{C}_1\\ \mathbf{C}_2\\ \mathbf{C}_3\end{bmatrix} = (c_{ij})_{i=1,2,3,\ j=1,2,3} = \mathbf{J}[\mathbf{T}_1^{(3)}\mathbf{r}_E^{(3)}, \dots, \mathbf{T}_3^{(3)}\mathbf{r}_E^{(3)}]\\ \mathbf{d} &= (d_i)_{i=1,2,3} = \mathbf{J}\Bigg(\Bigg(\sum_{i=1}^{n_{dof}}\sum_{j=1}^{n_{dof}} \mathbf{T}_{ij}^{(3)}\dot{q}_i\dot{q}_j\Bigg)\mathbf{r}_E^{(3)}\Bigg)\\ \mathbf{u} &= (u_j)_{j=1,2,3} = \frac{1}{a_{E,a}^{(0)})^2}\mathbf{J}_1\mathbf{T}^{(3)}\mathbf{r}_E^{(3)}\mathbf{C}_1 + \frac{1}{(b_{E,a}^{(0)})^2}\mathbf{J}_2\mathbf{T}^{(3)}\mathbf{r}_E^{(3)}\mathbf{C}_2\\ v &= \frac{1}{(a_{E,a}^{(0)})^2}[(\mathbf{C}_1\dot{\mathbf{q}})^2 + \mathbf{J}_1\mathbf{T}^{(3)}\mathbf{r}_E^{(3)}d_1] + \frac{1}{(b_{E,a}^{(0)})^2}[(\mathbf{C}_2\dot{\mathbf{q}})^2 + \mathbf{J}_2\mathbf{T}^{(3)}\mathbf{r}_E^{(3)}d_2]\end{aligned}$$

Next, the procedure requires coordinate derivatives partition according to $(2.5)_1$. We need to select independent, i.e. control inputs, and dependent velocities as follows

$$\begin{aligned}&i_{i_c} \in \{1\} \rightarrow \mathbf{q}_{i_c} = [(\psi^{(1)})]^{\mathrm{T}}\\ &i_{d_c} \in \{2,3\} \rightarrow \mathbf{q}_{d_c} = [(\psi^{(2)}, z^{(3)}]^{\mathrm{T}}\end{aligned} \tag{4.4}$$

Relation between dependent and independent velocities can be presented as

$$\dot{\mathbf{q}}_{d_c} = -\mathbf{K}_{d_c}^{-1}\mathbf{K}_{i_c}\dot{\mathbf{q}}_{i_c} \;\Rightarrow\; \frac{\partial\dot{\mathbf{q}}_{d_c}}{\partial\dot{\mathbf{q}}_{i_c}} = -\mathbf{K}_{d_c}^{-1}\mathbf{K}_{i_c} \tag{4.5}$$

where

$$\mathbf{K}_{d_c} = \begin{bmatrix}u_2 & u_3\\ c_{12} & c_{13}\end{bmatrix} \qquad \mathbf{K}_{i_c} = \begin{bmatrix}u_1\\ c_{11}\end{bmatrix}$$

Notice, that this partition reflects partition of the coordinates into the control inputs and the controlled ones. As a consequence, in the adopted manipulator model, it is assumed that the column is operated by a flexible drive. The driving torque is determined by the following formula

$$t_{dr}^{(1)} = -s_{dr}^{(1)}(\psi_{dr}^{(1)} - \psi^{(1)}) - d_{dr}^{(1)}(\dot{\psi}_{dr}^{(1)} - \dot{\psi}^{(1)}) \tag{4.6}$$

where $s_{dr}^{(1)}$ and $d_{dr}^{(1)}$ are stiffness and damping coefficients of the flexible drive, respectively.

The displacement of the column $\psi_{dr}^{(1)}$ is a function of time, and it is presented in Fig. 4.

It is assumed that friction is present in manipulator joints. The LuGre friction model (Armstrong-Hélouvry, 1991) is taken into account to calculate the friction coefficients

$$\mu^{(i)} = \sigma_0^{(i)}z^{(i)} + \sigma_1^{(i)}\dot{z}^{(i)} + \sigma_2^{(i)}\dot{q}_i \tag{4.7}$$

where $\sigma_0^{(i)}$, $\sigma_1^{(i)}$, $\sigma_2^{(i)}$ are stiffness, damping and viscous damping coefficients of the bristle, respectively, $z^{(i)}$ is the deflection of the bristle.

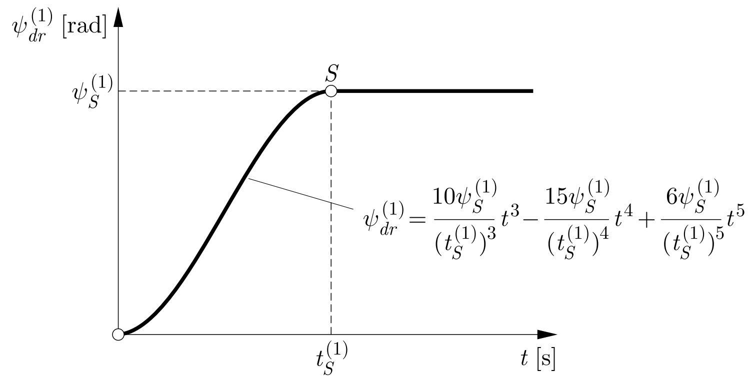


Fig. 4. Assumed displacement of the manipulator column

The deflection velocity $\dot{z}^{(i)}$ is calculated according to the formula

$$\dot{z}^{(i)} = \dot{q}_i - \frac{\sigma_0^{(i)} z^{(i)} \dot{q}_i \operatorname{sgn}(\dot{q}_i)}{\mu_k^{(i)} + (\mu_s^{(i)} - \mu_k^{(i)}) \exp\Big(-\Big(\frac{\dot{q}_i}{\dot{q}_{S,i}}\Big)^2\Big)} \tag{4.8}$$

where $\mu_s^{(i)}$, $\mu_k^{(i)}$ are static and kinetic friction coefficients, respectively, $\dot{q}_{S,i}$ is the Stribeck velocity.

The reference dynamic model of the manipulator is derived based upon modified GPME (2.4), i.e.

$$\begin{bmatrix} \mathbf{M}_i\Big|_{i\in i_{i_c}} + \sum\limits_{j\in i_{d_c}} \mathbf{M}_j \dfrac{\partial \dot{q}_j}{\partial \dot{q}_i} \\ \mathbf{u} \\ \mathbf{C}_3 \end{bmatrix} \ddot{\mathbf{q}}$$

$$= \begin{bmatrix} \mathbf{h}_i - \mathbf{g}_i + \mathbf{Q}_i + \sum\limits_{j\in i_{i_c}\cup i_{d_c}} \dot{q}_j \dfrac{\partial Q_j}{\partial \dot{q}_i} + \sum\limits_{k\in i_{d_c}} \Big(h_k + Q_k - g_k + \sum\limits_{j\in i_{i_c}\cup i_{d_c}} \dot{q}_j \dfrac{\partial Q_j}{\partial \dot{q}_k}\Big) \dfrac{\partial \dot{q}_k}{\partial \dot{q}_i} \\ -v - 2\alpha_1 \mathbf{u}\dot{\mathbf{q}} - \beta_1^2 \Big[\Big(\dfrac{x_E^{(0)}}{a_{E,a}^{(0)}}\Big)^2 + \Big(\dfrac{y_E^{(0)}}{b_{E,a}^{(0)}}\Big)^2 - 1\Big] \\ -d_3 - 2\alpha_2 \mathbf{C}_3\dot{\mathbf{q}} - \beta_2^2 (z_E^{(0)} - z_{E,a}^{(0)}) \end{bmatrix} \tag{4.9}$$

where α_i, $\beta_i\big|_{i=1,2}$ are coefficients of the Baumgarte numerical solution stabilization method.

5. Numerical studies – simulation results

Manipulator reference dynamic model (4.9) as well as control dynamics (3.2) and (3.3) are analysed in this Section. Specifically, we demonstrate the manipulator model behaviour when the programmed constraints are imposed. Parameters of the manipulator model are presented in Table 1.

Additionally, in the numerical calculations, the following parameters are assumed:

- Flexible drive data: $s_{dr}^{(1)} = 10^4$ Nm/rad, $d_{dr}^{(1)} = 70$ Nms/rad, $\psi_S^{(1)} = 3600°$, $t_S^{(1)} = 5$ ms;
- PD controller data: $\delta_1 = 40$, $\delta_2 = 3$, $\delta_3 = 60$;
- Runge-Kutta IV-order scheme: $h = 10^{-3}$ s;
- Baumgarte's coefficients stabilizing numerical solutions: $\alpha = 100$, $\beta = 50$.

Reference dynamics for four initial positions of link 3 with respect to horizontal x asis are analyzed, i.e. 45°, 60°, 70° and 80° are calculated using the GMPE algorithm. The reference

Table 1. Initial manipulator configuration and friction parameters

Parameter	Symbol	Link 1	Link 2	Link 3
Initial configuration	$q_i\|_{t=0}$	0	$270° - \alpha$	0.5 m
Static friction coefficient	$\mu_s^{(i)}$	0.1	0.1	0.1
Kinetic friction coefficient	$\mu_k^{(i)}$	0.2	0.2	0.2
Stiffness coefficient of bristle	$\sigma_0^{(i)}$	5.0 Nm/rad	5.0 Nm/rad	1.0 N/m
Damping coefficient of bristle	$\sigma_1^{(i)}$	0.025 Nm/rad	0.025 Nm/rad	0.02 Ns/m
Viscous damping coefficient	$\sigma_2^{(i)}$	0	0	0
Stribeck velocity	$\dot{q}_{S,i}$	0.175 rad/s	0.175 rad/s	0.001 m/s

time courses thus obtained are applied to the PD controller for the trajectory tracking problem. In numerical simulations, the influence of disturbances of the manipulator initial configuration on the system response and time courses of controls are analyzed. Figure 5a presents the trajectory of the end-effector E in the $x^{(0)}y^{(0)}$ plane of the reference frame $\{0\}$ for different initial configurations. The time course of $z_E^{(0)}$ coordinate is shown in Fig. 5b.

It can be seen that the change of the initial configuration of the manipulator has influence on the position of the effector in the initial moment and on the time after which the given programmed constraints will be achieved. The proposed control algorithm is also effective in the case of disturbances of the initial conditions. In Fig. 6, time courses of the joint coordinates are presented. It can be seen that the motion planner, i.e. the reference motion analysis, enables verification of the programmed motion planned for variety of constraint options. Also, changes in the initial configuration do not have a significant effect on motion of links 1 and 3, while the effect is significantly noticeable in the case of motion of rotary arm 2.

Absolute errors between the reference time courses of displacements and those obtained for the disturbed system, shown in Fig. 7, are determined as

$$\Delta_j^{(\alpha)}(t_i)\Big|_{\alpha\in 60°,70°,80°} = |q_j^{(\alpha)}(t_i) - q_j^{(ref)}(t_i)| \tag{5.1}$$

where $q_j^{(\alpha)}$ are values of the generalized coordinates obtained for angle α, $q_j^{(ref)}$ are reference values of generalized coordinates obtained for angle $\alpha = 45°$.

The results demonstrate that the greatest absolute errors occur in the case of motion of link 2. In the case of other links, they are negligible. It can also be noticed that after 2.5 s, motion of the system is consistent with the assumed constraints. In Fig. 8, time courses of torques and forces obtained from the PD controller are presented.

It can be seen that the torque acting on link 2 has the greatest values up to 1 s, which is related to compensation of disturbances caused by the initial conditions. The time courses of torques and forces have relatively high values, which is caused by the implementation of selected programmed constraints excusively for illustraiting the purposes.

6. Conclusions

Development of a complementary motion planning strategy for task-based motions and a control strategy architecture for tracking these motions based upon a computational procedure for generation of dynamical models with position and first order programmed constraints are presented in the paper. The procedure which was modified comparing to its generic version developed with the aid of the analytical dynamics approach, offers efficient generation of constrained dynamical models which are equivalent to reference dynamic describing motion under the constraints put

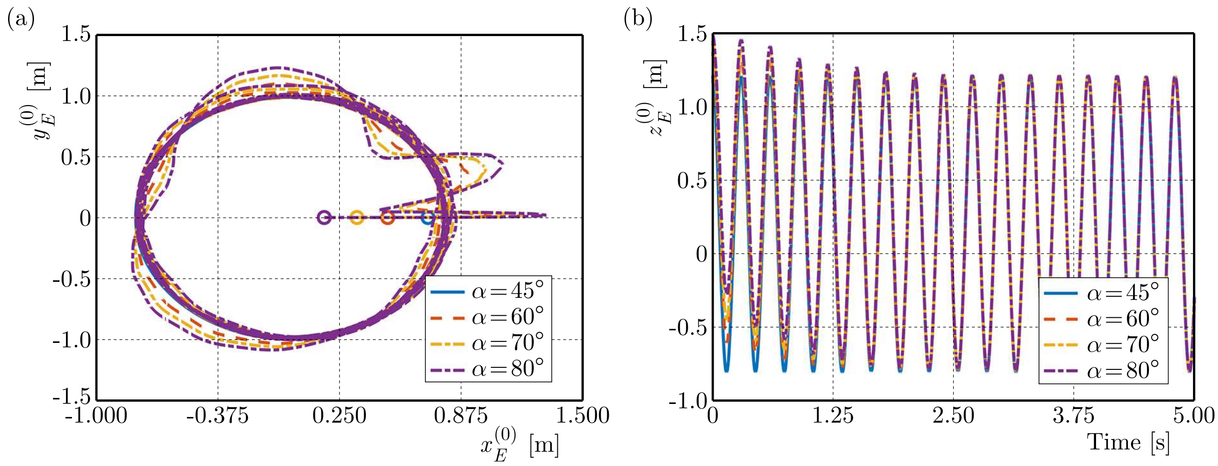


Fig. 5. (a) Trajectory of the end-effector E in the $x^{(0)}y^{(0)}$ plane; (b) time course of $z_E^{(0)}$ coordinate

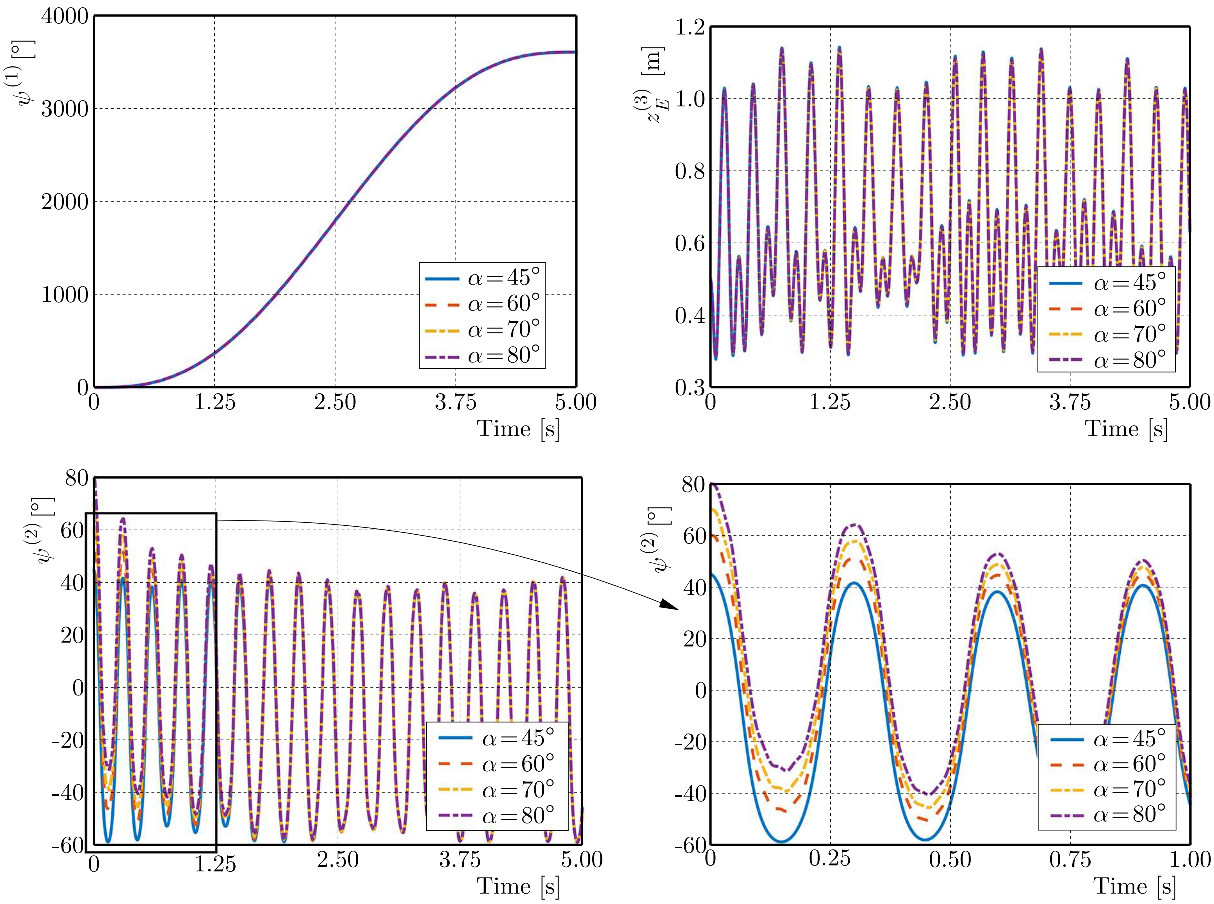


Fig. 6. Time course of joint coordinates

upon the system model. It is not based upon the Newton-Euler or Lagrange approaches but upon the GPME method. It provides reference dynamical models, which may serve as motion planners of constrained motions and enable insight into system performance under the constraints. In the presented procedure, the constraints may be material or nonmaterial, e.g. task-based, and the final equations of motion are derived in the reduced state form, i.e. constraint reaction forces are eliminated at the equations derivation process. This is the essential advantage of our approach comparing to existing dynamical modeling approaches. The effectiveness of this procedure was demonstrated by simulation studies of constrained motion performance of the three link manipulator model.

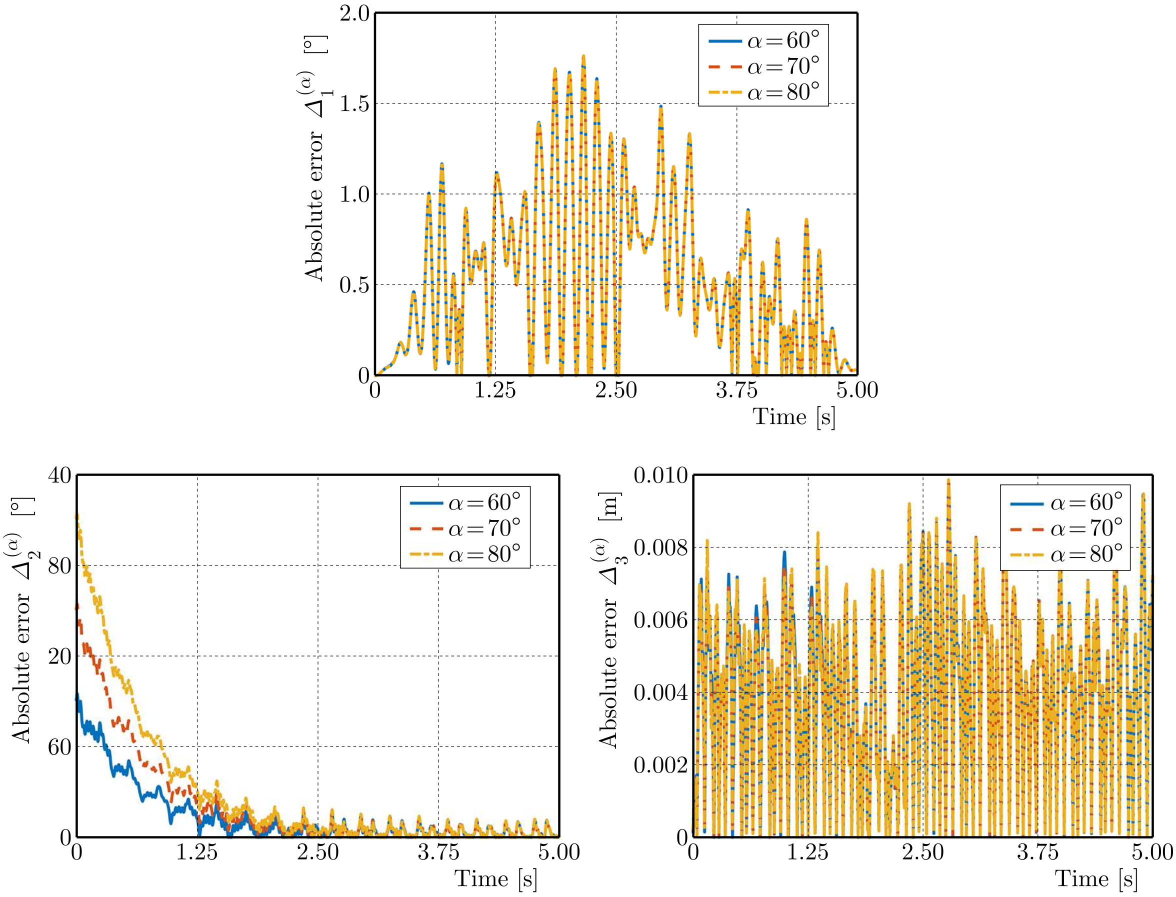


Fig. 7. Time courses of absolute errors for joint coordinates

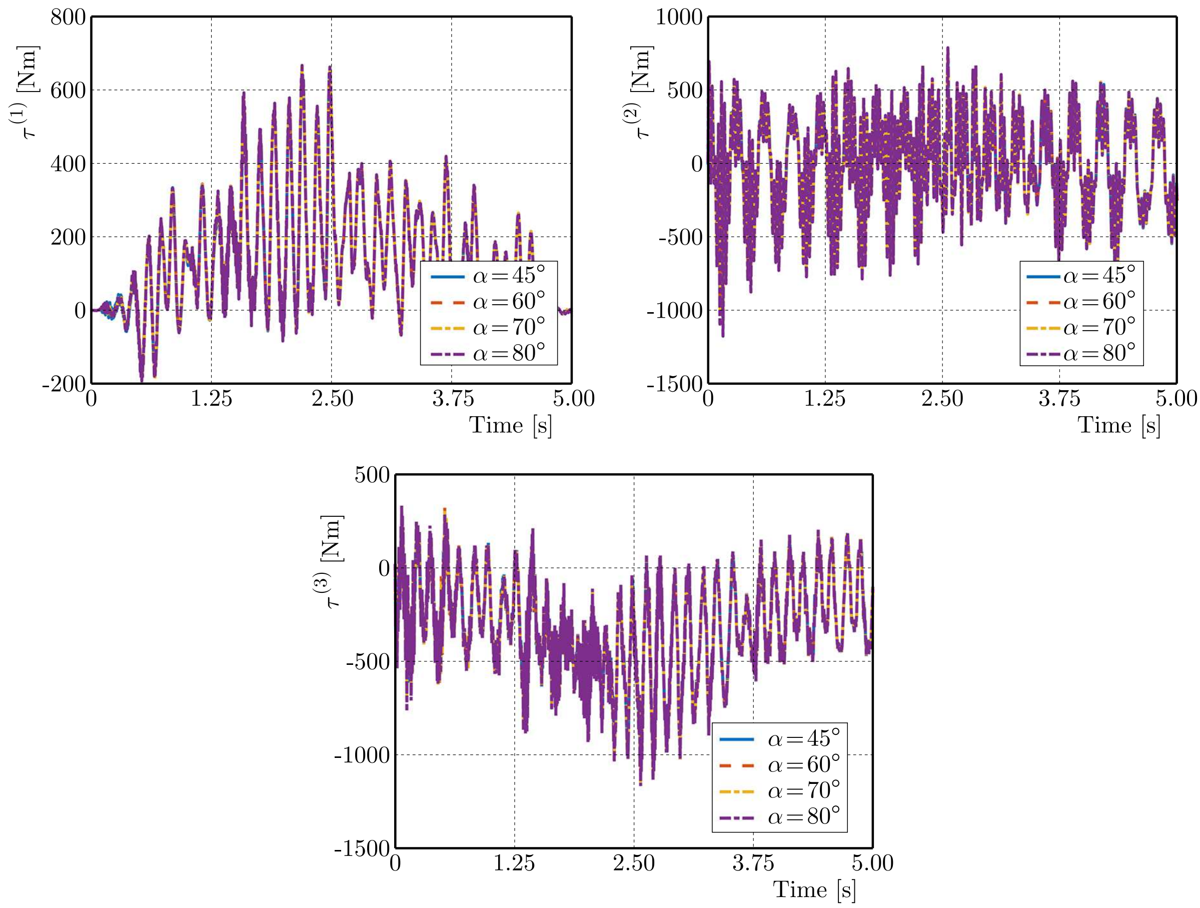


Fig. 8. Time courses of driving torques and the force

References

1. Armstrong-Hlouvry B., 1991, *Control of Machines with Friction*, Springer International Series in Engineering and Computer Science, New York

2. Banaszuk A., Mehta P.G, Hagen G., 2007, The role of control in design: From fixing problems to the design of dynamics, *Control Engineering Practice*, **15**, 10, 1292-1305

3. Bounini F., Gingras D., Pollart H., Gruyer D., 2017, Modified artificial potential field method for online path planning applications, *IEEE Intelligent Vehicles Symposium*, (IV), LA, CA, USA

4. Chiang H.-T., Malone N., Lesser K., Oishi M., Tapia L., 2015, Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments, *IEEE International Conference on Robotics and Automation (ICRA)*, Washington, USA

5. Dixon W.E., Behal A., Dawson D.M., Nagarkatti S.P., 2003, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*, Springer

6. Do T.-T., Vu V.-H., Liu Z., 2021, Symbolic differentiation algorithm for inverse dynamics of serial robots with flexible joints, *ASME Journal of Mechanisms and Robotics*, **13**, 6

7. Feyzabadi S., Carpin S., 2016, Multi-objective planning with multiple high level task specifications, *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden

8. Huang Y., van Gelder J.M., Wen Y., 2018, Velocity obstacle algorithms for collision prevention at sea, *Ocean Engineering*, **151**, 308-321

9. Jarzebowska E., 2008, Advanced programmed motion tracking control of nonholonomic mechanical systems, *IEEE Transactions on Robotics*, **24**, 6, 1315-1328

10. Jarzębowska E., 2009, Quasi-coordinates based dynamics modeling and control design for nonholonomic systems, *Nonlinear Analysis: Theory, Methods and Applications*, **71**, 12, 118-131

11. Jarzębowska E., 2012, *Model-Based Tracking Control of Nonlinear Systems*, Boca Raton: CRC Press, Taylor & Francis Group

12. Jarzębowska E., Augustynek K., Urbaś A., 2018a. Automated generation of reference dynamical models for constrained robotic systems in the presence of friction and damping effects, *Concurrency and Computation: Practice and Experience*, 4452

13. Jarzębowska E., Augustynek K., Urbaś A., 2018b, Development of a computational based reference dynamics model of a flexible link manipulator, *Springer Proceedings Mathematics and Statistics Series, Dynamical Systems in Applications*, **249**, 169-180

14. Jarzębowska E., Augustynek K., Urbaś A., 2023, Dynamics modeling method dedicated to system models with open- and closed-loop structures subjected to kinematic and task based constraints, *Nonlinear Dynamics*, **111**, 13, 12053-12080

15. Jarzębowska E., Cichowski M., 2018, Dynamics modeling and performance analysis of underwater vehicles based on the Boltzmann-Hamel equations approach, *MATEC Web of Conferences*, **148**, 03005

16. Kane T.R., Levinson D.L., 1983, The use of Kane's dynamical equations in robotics, *International Journal of Robotics Research*, **2**, 3, 3-21

17. Khalil W., Boyer F., Morsli F., 2017, General dynamic algorithm for floating base tree structure robots with flexible joints and links, *ASME Journal of Mechanisms and Robotics*, **9**, 3

18. LaValle S., 2006, *Planning Algorithms*, Cambridge University, Cambridge

19. Lee B.H., Jeon J.D., Oh J.H., 2017, Velocity obstacle based local collision avoidance for a holonomic elliptic robot, *Autonomous Robots*, **41**, 6, 1347-1363

20. Liu J., Liu R., 2016, Simple method to the dynamic modeling of industrial robot subject to constraint, *Advances in Mechanical Engineering*, **8**, 4, 1-9

21. Lu Y., Xi Z., Lien J.-M., 2016, Online collision prediction among 2D polygonal and articulated obstacles, *The International Journal of Robotics Research*, **35**, 5, 476-500
22. Macfarlane S., Croft E.A., 2003, Jerk bounded manipulator trajectory planning: Design for real-time applications, *IEEE Transactions on Robotics and Automation*, **19**, 1, 42-52
23. Sayers M.W., 1990, Symbolic computer methods to automatically formulate vehicle simulation codes, Ph.D. Thesis, Dept. of Mechanical Engineering, University of Michigan
24. Van den Berg J., Overmars M., 2008, Planning time-minimal safe paths amidst unpredictably moving obstacles, *International Journal of Robotics Research*, **27**, 11-12, 1274-1294
25. Yang X., Zhang X., Xu S., Ding Y., Zhu K., Liu X., 2019, An approach to the dynamics and control of uncertain robot manipulators, *Algorithms*, **12**, 3

Manuscript received October 26, 2023; accepted for print February 21, 2024