2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Integrated Time-Optimal Trajectory Planning and Control Design for Industrial Robot Manipulator

Stepan S. Pchelkin, Anton S. Shiriaev, Anders Robertsson, Leonid B. Freidovich

Abstract— **We consider planning and implementation of fast motions for industrial manipulators constrained to a given geometric path. With such a problem formulation, which is quite reasonable for many standard operation scenarios, it is intuitively clear that a feedback controller should be designed to achieve orbital stabilization of a time-optimal trajectory instead asymptotic. We propose an algorithm to convert an asymptotically stabilizing controller into an orbitally stabilizing one and check achievable performance in simulations and, more importantly, in experiments performed on a standard industrial robot ABB IRB 140 with the IRC5-system extended with an open control interface. It is verified that the proposed re-design allows significantly reduced deviations of the actual trajectories from the desired one at high speeds not only for a chosen base feedback design but also outperforming the state-of-the-art commercial implementation offered by ABB Robotics.**

Index Terms— **Motion and trajectory planning with constraints, Orbital stabilization, Time optimal control, Industrial manipulators**

I. Introduction

Robotic systems with higher levels of autonomy are to become key elements of advanced factories of the future. Most of robotic manipulation tasks require the deviations of the executed motion from a nominal one, preplanned according to a scenario, to be as small as possible (and persistently repeatable). Such nominal motions are often chosen to be the best with respect to one or several optimality criteria and not to violate specifications imposed by the task. For instance, if in a scenario the end-effector of a robot should follow a particular geometrical path in a configuration space with predefined path-dependent sequence of orientations, it is common to plan a time-(sub)optimal trajectory or its approximation, which is consistent with velocity and acceleration constraints imposed by various kinematic and dynamic limitations. Since the middle of the 1980s, this constraint-based optimization for velocity assignment has become in focus of the robotics research community and upon various assumptions solutions were found, see e.g. [1], [2], [3], [4] and references therein.

However, the implementation and realization for the found time-optimal trajectories for industrial robots manipulators in experiment is a challenging task. Indeed, the phase-plane arguments of [2] are model-based and, therefore, might be sensitive to unavoidable uncertainties in parameters of the model of the robot, to presence of excited hidden dynamics of non-collocated compliances of joints that are non-negligible for an aggressive (optimal!) motion, as well as to mismatches in estimates for payload mass-distribution, dimensions, *etc.* As a result, even the best available commercial software packages for motion planning, such as [5], and motion control might not always guarantee the required accuracy in generating such motions in experiments.

In this paper, we present control re-design arguments, integrated with a motion planning strategy, that can improve robustness, accuracy and repeatability of time-optimal trajectories. In this way, we have complemented the classical and well-known motion planning methods bringing efforts for an appropriate formulation of control task and arguments for control design solutions and their implementation. To illustrate the contribution, we have chosen to work with an ABB IRB 140 robot. Below we are to re-iterate and to apply the known motion planning arguments for one of the targeted behaviors, and to elaborate in detail a new control re-design steps and then to test performance of the closed-loop system in simulation and in real hardware experiments.

II. Problem Formulation and Preliminaries

A. Modeling the Manipulator

Let us concentrate on manipulation tasks related to moving the end effector from a start point to an end point in the world frame, i.e. local grasping tasks are not considered here. Thus, the robot geometry to be described is an open kinematic chain of six joints from the base to the links where the end effector is attached. In Fig. 1 a schematic picture of the ABB IRB 140 robot manipulator with 6 Degrees of Freedom (DOF) is shown.

However, for simplicity, below we consider only 3 DOF of the robot corresponding to axis 1, axis 2, and axis 3 (see Fig. 1). The joint variables (angles) form the vector of generalized coordinates $q = [q_1, q_2, q_3]^T$ for these 3 DOF.

The forward kinematics can be conveniently expressed using the Denavit-Hartenberg (DH) convention [7], where configuration of each (i-th) link is represented by a homogeneous transformation

$$A_i(q_i) = Rot_{z,q_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha}, \quad i = 1, 2, 3 \tag{1}$$

formed by standard elementary translations and rotations, see

S. Pchelkin and A. Shiriaev are with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway. E-mail: Stepan.Pchelkin@itk.ntnu.no.

L. Freidovich and A. Shiriaev are with the Department of Applied Physics and Electronics, Umeå University, SE-901 87 Umeå, Sweden.

A. Robertsson is with the Dept. of Automatic Control, Lund University, SE-221 00 Lund, Sweden and is member of the LCCC Linnaeus Center and the ELLIIT Excellence Center at Lund University.

This work is part of the project 'Next Generation Robotics for Norwegian Industry'. The project is funded in part by the Norwegian Research Council, Statoil, Hydro, Tronrud Engineering, Glen Dimplex Nordic, RobotNorge.

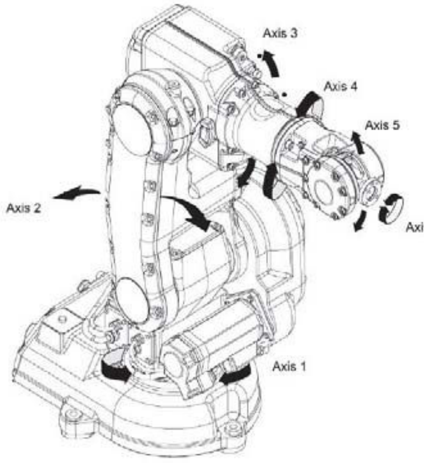

Fig. 1. The robot manipulator with 6 DOF [6].

e.g. [7], and parameterized by the joint angle q_i, the link offset d_i, the link length a_i, and the link twist α.

The Cartesian position of the flange with respect to the base frame of the robot is defined by

$$P^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 1} \end{bmatrix} T_3^0 \begin{bmatrix} 0_{3\times 1} \\ 1 \end{bmatrix} \quad (2)$$

where $I_{3\times 3}$ is the identity matrix, $0_{3\times 1}$ is a zero vector, and $T_3^0 = A_1(q_1)A_2(q_2)A_3(q_3)$.

Inverse kinematics from a configuration of the flange to the joint variables can be found as a solution of a set of nonlinear trigonometric equations given by T_3^0 in (2).

The configuration space of the manipulator is spanned by the joint variables and it is restricted by the mechanical construction of the robot. Differential constraints, on the other hand, are imposed by the dynamics of the system, which, to some extent, can be described by the following differential equation

$$M(q)\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + C(q,\dot{q})\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + G(q) = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (3)$$

where $\tau = [\tau_1, \tau_2, \tau_3]^T$ is the vector of control torques applied at the joints, $M(q)$ is the inertia matrix, the vector $C(q,\dot{q})\dot{q}$ represents Coriolis and centrifugal generalized forces, and $G(q)$ is due to gravity.

For a robot manipulator, velocity constraints show up naturally due to limited rotation speeds of the actuators, whereas acceleration constraints are defined by maximal achievable torques of the actuators. In this study, we use configuration and velocity constraints given in Table. I. These ranges of allowable velocities and angles are taken from the robot manual.

TABLE I
POSITION AND VELOCITY CONSTRAINTS FOR CONSIDERED LINKS OF THE MANIPULATOR

Link i	$q_{i,min}$	$q_{i,max}$	$\dot{q}_{i,min}$	$\dot{q}_{i,max}$
1	-3.1415 rad	3.1415 rad	-3.4907 rad/s	3.4907 rad/s
2	-1.5708 rad	1.9199 rad	-3.4907 rad/s	3.4907 rad/s
3	-4.0143 rad	0.8727 rad	-4.5379 rad/s	4.5379 rad/s

B. Path Planning

Let us specify a path that the flange should follow over time. In this study, we are not concerned with the path-planning problem. So, we concentrate on the goal of computing a feasible fast time-evolution along a predefined path subject to differential constraints.

Let us introduce a sample desired path that we are going to work with from here on. This path describes a 2D circle in the $x-y$ plane of the base frame as depicted in Fig. 2.

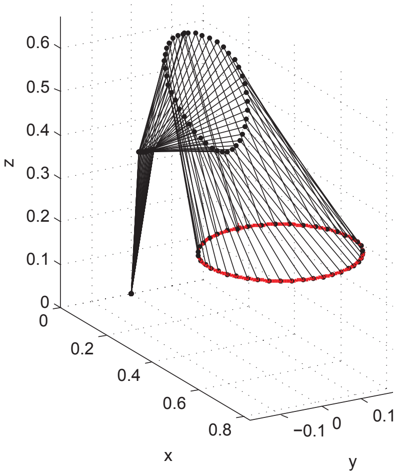

Fig. 2. Path of the end effector of the robot describing a circle in the $x-y$ plane.

This path in the task space can be parameterized as follows

$$P^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos\theta + x_0 \\ r\sin\theta \\ z_0 \end{bmatrix} \quad (4)$$

where $r = 0.2$, $x_0 = 0.65$, $z_0 = 0.3$, and $\theta \in [-\pi, \pi]$.

Solving the inverse kinematics problem, one can define the desired path in the joint space

$$q = \begin{bmatrix} q_1^*(x,y,z) \\ q_2^*(x,y,z) \\ q_3^*(x,y,z) \end{bmatrix} \quad (5)$$

In Fig. 3 below, q_1^*, q_2^*, and q_3^* are shown as functions of the parameter θ, which is taken as the angle defining the location along the circle.

C. Path-Constrained Trajectory Planning

Let us illustrate a parameterization of motions along predefined paths without explicit dependence on time. The procedure is exemplified for the circular path (4).

At first, we define a new variable that describes the path as a function of the generalized coordinates. For instance, the arc length along the path would be one choice that naturally

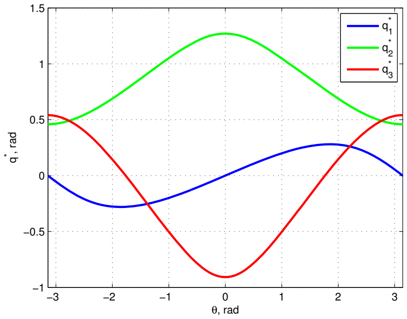

Fig. 3. Evolution of coordinates q_1^* (blue line), q_2^*, (green line) q_3^* (red line) for the target trajectory as functions of $\theta \in [-\pi, \pi]$.

yields a monotonic evolution. For our case of the circular path (4), it is also natural to take the angular position as such a path coordinate

$$\theta = atan2(y, x - x_0) \tag{6}$$

associated with a point P^0 on the circle (see Fig. 2).

As the next step, all the target evolutions of the joint variables $q = q^*(t)$ must be parameterized as functions of the new variable θ instead of time

$$q = \begin{bmatrix} q_1^*(t) \\ q_2^*(t) \\ q_3^*(t) \end{bmatrix} = \Phi(\theta)|_{\theta=\theta^*(t)} = \begin{bmatrix} \phi_1(\theta) \\ \phi_2(\theta) \\ \phi_3(\theta) \end{bmatrix}\Bigg|_{\theta=\theta^*(t)} \tag{7}$$

This can be interpreted as a synchronization of all joints along the path clocked to an independent configuration variable. With such representation the explicit dependence of time disappears and, thus, $\theta = \theta^*(t)$ can be viewed as a motion generator. Once a velocity profile for θ is chosen, all joint velocities are directly assigned by

$$\dot{q} = \Phi'(\theta)\dot{\theta} \tag{8}$$

It means that the nominal evolution of the full state space vector $[q, \dot{q}]^T$ is parameterized along the path, without even using the system dynamics (3). This approach is known as path-constrained trajectory planning [8], which is subject to velocity and acceleration constraints. In the context of control theory, the geometric function (7) is called a virtual holonomic constraint [9], [10], if it is preserved by some control action along solutions of the closed-loop system.

Finally, we have to assign a velocity profile along the path taking into account that there are configuration-dependent differential constraints. In this study only velocity constraints (see Table I) shall be considered. It is straightforward to also account for acceleration constraints linked to the system dynamics (3) as soon as a reliable quantification of them is obtained. However, we have found out that they are not as restrictive as the velocity constraints and so they are not to be taken into account.

Individual joint velocity and acceleration constraints can be efficiently handled introducing a differential equation relating θ and $\dot{\theta}$. The time evolution of θ along the path may be derived by integrating this differential equation

$$\dot{\theta} = h(\theta) \tag{9}$$

which consequently defines the time evolution of all the joint coordinates in (7).

We introduce the function path coordinate θ^* so that the velocity along the trajectory is the maximum possible:

$$\begin{aligned} \dot{s}_i &= \max\left(\left|\frac{\dot{q}_{i,max}}{\phi_i'}\right|, \left|\frac{\dot{q}_{i,min}}{\phi_i'}\right|\right) \\ \left|\dot{\theta}^*\right| &\leq \min\left(\dot{s}_i\right) \end{aligned} \tag{10}$$

where $\dot{q}_{i,min}$, $\dot{q}_{i,max}$ are physical constraints for the ith joint velocity (see Table I). We choose the function in the right-hand side of (9) as a Bezier polynomial of order 10.

From the differential equation (10), we obtain velocity profiles along the path in the $(\theta, \dot{\theta})$-plane, see Fig. 4. In Fig. 5, the obtained time-evolution of θ^* is shown. From Fig. 5, we can see that the period of motion is $T = 0.92$ (s) for our choice of a sub-optimal $\theta^*(t)$. Note that this is faster than the trajectory generated by the standard ABB planner run with the "maximal velocity" settings.

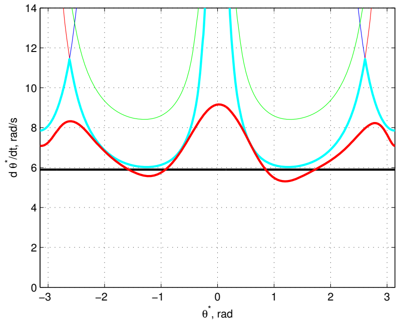

Fig. 4. Velocity profiles along the path in the $(\theta,\dot{\theta})$-phase plane (red line). The path-constrained optimal curve yields a faster motion compared with the driver. Velocity constraints along the path are above the cyan line. Velocities at the start and end are not zero and equal.

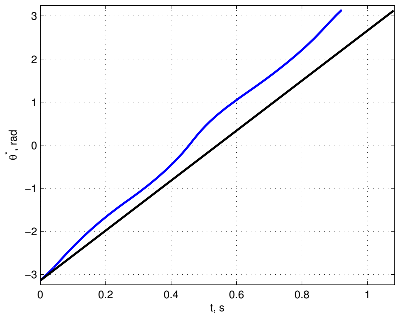

Fig. 5. Obtained function θ^* is blue line. Time evolution θ for the ABB generated trajectory is presented by the black line.

D. Control Design for Asymptotic Stability

Motion planning should be complimented with a feedback control design. A possible choice is the controller originally introduced in [11]. It was one of the first results guaranteeing tracking for rigid-joint robots achieving global uniform asymptotic stability (GUAS) of the origin $(q - q^*, \dot{q} - \dot{q}^*) = (0, 0)$ for the model (3). It is a natural extension of the famous tracking controller in the form of PD (proportional-derivative) feedback plus gravity cancellation, suggested for the regulation problem in [12].

The PD+ control law $\tau = U(e, \dot{e}, Q^*(t))$ of [11] is

$$\tau = \hat{M}(q)\ddot{q}^*(t) + \hat{C}(q, \dot{q})\dot{q}^*(t) + \hat{G}(q) - K_p\, e - K_d\, \dot{e} \quad (11)$$

where

$$\begin{aligned} e = q - q^*(t), \qquad \dot{e} = \dot{q} - \dot{q}^*(t), \\ Q^*(t) = [q^*(t), \dot{q}^*(t), \ddot{q}^*(t)] \end{aligned} \quad (12)$$

K_d, K_p are (diagonal) positive definite matrices; $\hat{M}(q)$, $\hat{C}(q, \dot{q})$, and $\hat{G}(q)$ are available estimates for $M(q)$, $C(q, \dot{q})$, and $G(q)$ respectively. Note that for implementation of this controller $q^*(t)$ and $\dot{q}^*(t)$ must be specified as functions of time and be available on-line.

We will show below how to transform this regulator into a one that is appropriate for orbital stabilization. After that we are to investigate performance of the closed-loop system.

III. Main Result

A. Re-design of Feedback Control

Suppose we are given an asymptotically stabilizing feedback control law

$$\tau = U(e, \dot{e}, Q^*(t)), \quad (13)$$

with (12), where $q^*(t)$ is defined by (7) and $\theta = \theta^*(t)$ being a solution of the differential equation (9), right-hand side of which is derived from the sub-optimal motion planning procedure as above.

We are interested here to propose a procedure to transform it into another feedback control law that ensure asymptotic orbital stability. Roughly speaking, in essence, instead of classical stability: $\forall \varepsilon > 0\ \exists \delta > 0$ such that

$$\left\| \begin{bmatrix} q(0) - q^*(0) \\ \dot{q}(0) - \dot{q}^*(0) \end{bmatrix} \right\| \le \delta \Rightarrow \left\| \begin{bmatrix} q(t) - q^*(t) \\ \dot{q}(t) - \dot{q}^*(t) \end{bmatrix} \right\| \le \varepsilon \quad \forall t \ge 0$$

we need orbital stability: $\forall \varepsilon > 0\ \exists \delta > 0$ such that

$$\begin{aligned} \inf_{s \in [0,T]} \left\| \begin{bmatrix} q(0) - q^*(s) \\ \dot{q}(0) - \dot{q}^*(s) \end{bmatrix} \right\| \le \delta \\ \Rightarrow \quad \inf_{s \in [0,T]} \left\| \begin{bmatrix} q(t) - q^*(s) \\ \dot{q}(t) - \dot{q}^*(s) \end{bmatrix} \right\| \le \varepsilon \quad \forall t \ge 0 \end{aligned}$$

where s is some variable that determines the shortest distance between the trajectories $q(t)$ and $q^*(t)$ and instead of ensuring asymptotic time-wise closeness

$$\lim_{t \to \infty} \left\| \begin{bmatrix} q(t) - q^*(t) \\ \dot{q}(t) - \dot{q}^*(t) \end{bmatrix} \right\| = 0$$

we would like to have asymptotic orbital closeness

$$\lim_{t \to \infty} \left\{ \inf_{s \in [0,T]} \left\| \begin{bmatrix} q(t) - q^*(s) \\ \dot{q}(t) - \dot{q}^*(s) \end{bmatrix} \right\| \right\} = 0$$

where T is the period of the planned target trajectory; see [13], [10] for related discussions.

Inspired by a natural choice of coordinates that are transversal to the target trajectory used in [10], we propose to substitute the classical tracking error $e = q - q^*(t)$ (and $\dot{e} = \dot{q} - \dot{q}^*$) measuring time-wise distance by the synchronization errors

$$\begin{aligned} y = q - \Phi(\theta), \qquad \dot{y} = \dot{q} - \Phi'(\theta)\,\dot{\theta}, \\ \theta = \mathrm{Pr}(q, \dot{q}), \qquad \dot{\theta} = h(\theta) \end{aligned} \quad (14)$$

where $\mathrm{Pr}(q, \dot{q})$ is an arbitrary smooth projection operator defined to ensure that $\mathrm{Pr}(q^*(t), \dot{q}^*(t)) = \theta^*(t)$.

In this way, we obtain

$$\tau = U(y, \dot{y}, Q^*(t)), \quad (15)$$

that should lead to asymptotic orbital stability.

B. Stability Statements

Let us first remind the result of [11].

Theorem 1: Suppose either $q^*(t)$ is smooth and periodic or Q^*, defined in (12), is uniformly bounded. Moreover, suppose $M(q)$ is uniformly bounded from below and above (as can be easily verified for our model). Consider the closed-loop system (3) and (13) and suppose there exists a positive definite in e and $\dot{e}$ function $V(e, \dot{e}, q^*(t), \dot{q}^*(t))$ such that

$$\tfrac{d}{dt} V(e, \dot{e}, q^*(t), \dot{q}^*(t)) \le - \begin{bmatrix} e \\ \dot{e} \end{bmatrix}^T Q_1(e, \dot{e}, Q^*(t)) \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$$

where either $Q_1(e, \dot{e}, Q^*(t))$ is positive definite in e and $\dot{e}$ or it is positive semi-definite but restricted to the set where $\frac{d}{dt} V(e, \dot{e}, q^*(t), \dot{q}^*(t)) = 0$ we have

$$\tfrac{d^3}{dt^3} V(e, \dot{e}, q^*(t), \dot{q}^*(t)) \le - \begin{bmatrix} e \\ \dot{e} \end{bmatrix}^T Q_3(e, \dot{e}, Q^*(t)) \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$$

where $Q_3(e, \dot{e}, Q^*(t))$ is positive definite in e and $\dot{e}$.

Then, $(e, \dot{e}) = (0, 0)$ is an asymptotically (uniformly) stable solution. ■

We should notice that under some mild additional technical assumptions global stability can be verified as well.

This statement was used in [11] to verify UGAS of (3) and (11) with the help of the natural Lyapunov function candidate

$$V(e, \dot{e}, q^*(t), \dot{q}^*(t)) = \tfrac{1}{2} \left(\begin{bmatrix} e \\ \dot{e} \end{bmatrix}^T M(q) \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + e^T K_p\, e \right)$$

that under assumptions $\hat{M} = M$, $\hat{C} = C$, $\hat{G} = G$ and C being taken so that the matrix $\dot{M}(q) - 2C(q, \dot{q})$ is skew-symmetric, leads to $\frac{d}{dt} V = -\dot{e}^T K_d\, \dot{e}$ and to UGAS.

The following statement is valid for our modification of the feedback control law.

Theorem 2: (Main result) Suppose either $q^*(t)$ is smooth and periodic or Q^*, defined in (12), is uniformly bounded. Moreover, suppose $M(q)$ is uniformly bounded from below and above. Suppose there exists a positive definite in e and $\dot{e}$ function $V(e, \dot{e}, q^*(t), \dot{q}^*(t))$ that satisfies conditions of Theorem 1.

Then, $(q, \dot{q}) = (q^*(t), \dot{q}^*(t))$ is an asymptotically orbitally stable solution of the closed-loop system (3), (15), (14). ■

The validity of Theorem 2 can be verified using a slight modification of the arguments presented in [11], with the help of a Lyapunov function candidate taken as $V(y, \dot{y}, q^*(t), \dot{q}^*(t))$.

Correspondingly, we obtain a new asymptotically orbitally stabilizing controller

$$\tau = \hat{M}(q)\ddot{q}^*(t) + \hat{C}(q, \dot{q})\dot{q}^*(t) + \hat{G}(q) - K_p y - K_d \dot{y} \quad (16)$$

while other controllers to achieve asymptotic orbital stability can be easily constructed in a similar manner.

Note, however, that to ensure global stability, additional properties of the projection function are required.

C. Simulation Results

Now we simulate the system (3) with controls (11) and (16). The simulation results of the system (3) with control (11) are showed in Figs. 6 - 7. We choose matrices for controller as $K_p = diag(250, 250, 250)$, $K_d = diag(50, 50, 50)$. Initial condition for simulation is $q(0) = q_d(0) + \epsilon_1$, $\dot{q}(0) = 0$, where ϵ_1 is an error in the initial position.

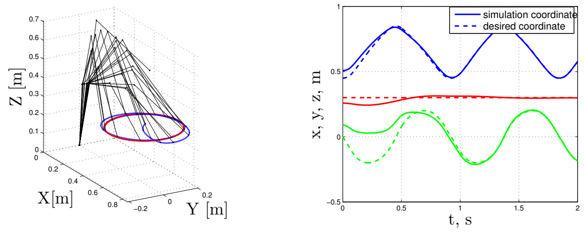

Fig. 6. *Left:* Resulting trajectory (blue line) achieved through simulation system (3) with control (11) in task space. Red line is the desired trajectory. *Right:* Resulting trajectory in joint space. Dashed lines are the desired trajectories $q^*(t)$ (q_1 – blue line, q_2 – green line, q_3 – red line).

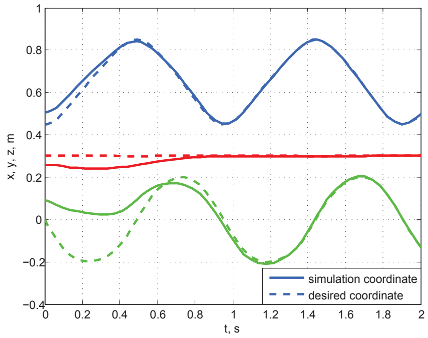

Fig. 7. Resulting trajectory in task space achieved through simulation system (3) with control (11). Dashed lines are desired trajectory $x(t)$ – blue line, $y(t)$ – green line, $z(t)$ – red line.

The result of the simulation is showed in Fig. 8 and Fig. 9. The matrices K_p and K_d are the same as in the previous simulation. Initial conditions for simulation are $q(0) = q_d(0) + \epsilon_1$ and $\dot{q}(0) = 0$, where ϵ_1 is an initial position error.

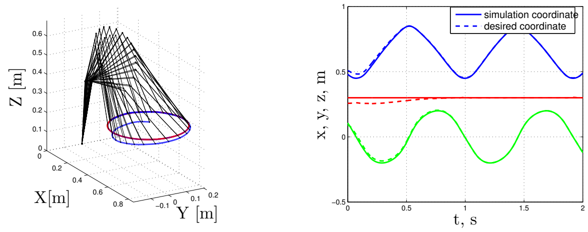

Fig. 8. *Left:* Resulting trajectory (blue line) achieved through simulation of system (3) with modified control (16) in task space. Red line is desired trajectory. *Right:* Resulting trajectory in joint space. Dashed lines are desired trajectory q^* (q_1 – blue line, q_2 – green line, q_3 – red line).

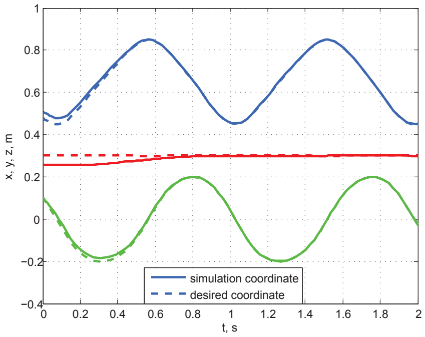

Fig. 9. Resulting trajectory in task space achieved through simulation of system (3) with control (16). Dashed lines are desired trajectory $x(t)$ – blue line, $y(t)$ – green line, $z(t)$ – red line.

By simulation results, we see that the system with control law (16) faster converges to the desired trajectory than with the control law (11).

D. Analysis of Robustness

Let us check the sensitivity and response of the closed-loop system to errors in the initial state of the robot and to uncertainties in the parameters. In particular, it is of interest to know whether the virtual holonomic constraints (vector of synchronization errors), which is specified in the path planning step, and defined by

$$Y = [y_1, y_2, y_3]$$

$$\begin{aligned} y_1 &= q_1 - \phi_1(P(q)) \\ y_2 &= q_2 - \phi_2(P(q)) \\ y_3 &= q_3 - \phi_3(P(q)) \end{aligned} \quad (17)$$

are kept small for the closed-loop system with uncertainties in parameters of the system (3). This is an important issue for implementation, since the sensitivity of the closed loop system dynamics might limit the use of the motion planner aimed at the time optimal performance. It is expected that the most sensitive parameters of the three links of model (3) would be parameters of the third link representing the most distant part of the mechanism from the base of the robot. For this we simulate the system with uncertainties in center of mass of the third link and mass of the third link. In Fig. 10 the results of simulation are shown. Analyzing

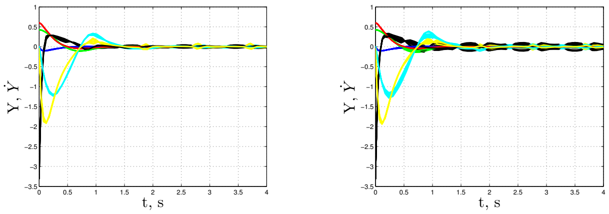

Fig. 10. *Left:* Evolution of coordinate Y with uncertainties in center of mass of the third link varying within the range of 10% of its nominal value. *Right:* Evolution of coordinate Y with uncertainties in mass of the third link varying within the range of 10% of its nominal value.

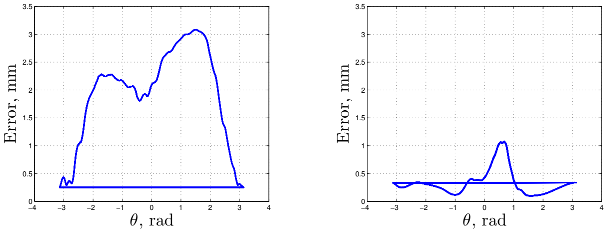

Fig. 11. *Left:* Error between desired and real trajectory of the robot for standard ABB controller for one period. *Right:* Error between desired and real trajectory of the robot for our controller for one period.

the simulation results, we can conclude that the closed-loop system with control law (16) is not sensitive to induced parametric uncertainty.

IV. Experimental Results

In this part of the paper we concentrate on the real-time implementation of the desired controls on IRB 140 robotic manipulator. The experiments are carried out with in real-time at a sampling frequency of $\Delta_s = 0.004s$. It is same as the sampling time of the standard ABB controller. We use parameters obtained through identification for the dynamic model in (3). The non-linear friction is compensated by a model-based addition to the control signal. Due to the facts that the robot components are mounted together and that their friction models are not known, an appropriate lumped model obtained by identification is used. In robots, friction significantly reduces performance and increase negative effects such as tracking errors, large settling times or limit cycles. In our control system we take into account such effects it in order to gain accuracy.

Here, we show results of experiment for the trajectory of Fig. 2 using control (16) and compare accuracy of following the trajectory using our controller and the controller provided by a standard commercial ABB system. In Fig. 11 errors between desired trajectory and trajectory of the robot for our controller and for the standard ABB controller are presented. The errors are calculated as the distance between the desired and the real trajectories. Path accuracy are $1.1mm$ for our controller and $3mm$ for the standard controller, respectively at the highest achievable velocity of the manipulator.

V. Conclucions

In this paper we have considered the problem of trajectory planning and control for an industrial manipulator. In trajectory planning we concentrated on the dominant velocity constraints which are apparent in actuators. It is straightforward to also account for acceleration constraints.

We have illustrated the case of a circular Cartesian path and two trajectories are chosen as examples. The angular position of a point on the path is used as an independent configuration variable for parameterization of the whole motion in terms of a virtual holonomic constraint and a velocity profile of the new variable along the motion. Such a parameterization can be instrumental and helps with assignments of velocity profiles along the same path so that we can optimize trajectories for execution time or other performance indices. We have planned trajectory with period $0.915s$. It is 25% faster than the trajectory generated by the standard ABB planner. The explicit dependence on time disappeared, which allows for implementation of our proposed control strategy, built upon standard tracking controllers, that achieves invariance of a preplanned trajectory and contraction to it if the system is moved away by external forces. Experimental tests demonstrate the feasibility of the proposed strategy. We can see that implementation of our design allows to obtain better accuracy than achievable by the currently commercially available standard controller.

References

[1] J. Hollerbach, “Dynamic scaling of manipulator trajectories,” *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 106, no. 1, pp. 102–106, 1984.

[2] J. Bobrow, S. Dubowsky, and J. Gibson, “Time-optimal control of robotic manipulators along specified paths,” *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.

[3] K. Shin and N. McKay, “Minimum-time control of robotic manipulators with geometric path constraints,” *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, 1985.

[4] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, “Time-optimal path tracking for robots: A convex optimization approach,” *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[5] ABB Robotics, *Operating manual RobotStudio 5.15*, document id: 3hac0232104-001 Revision: J ed. ABB, 2012.

[6] ——, *Product manual. IRB 140 Typ C; IRB 140T Typ C; IRB 140-6/0.8 Typ C; IRB 140T-6/0.8 Typ C*, document id: 3hac027400-001 Revision: E ed. ABB, 2004-2009.

[7] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. New York: John Wiley & Sons, 2006.

[8] S. LaValle, *Planning Algorithms*. New York: Cambridge University Press, 2006.

[9] A. Shiriaev, J. Perram, A. Robertsson, and A. Sandberg, “Periodic motion planning for virtually constrained Euler–Lagrange systems,” *Systems and Control Letters*, vol. 55, pp. 900–907, 2006.

[10] A. Shiriaev, L. Freidovich, and S. Gusev, “Transverse linearization for controlled mechanical systems with several passive degrees of freedom,” *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 893–906, 2010.

[11] B. Paden and R. Panja, “Globally asymptotically stable pd+ controller for robot manipulators,” *Int. J. of Contr.*, vol. 47, pp. 1697–1712, 1988.

[12] M. Takegaki and S. Arimoto, “A new feedback method for dynamic control of manipulators,” *Journal of Dynamic Syst., Meas., and Control – Trans of ASME*, vol. 103, no. 2, pp. 119–125, 1981.

[13] G. Leonov, “Generalization of the Andronov-Vitt theorem,” *Regular and Chaotic Dynamics*, vol. 11, no. 2, pp. 281–289, 2006.