Encoderless Position Control of a Two-Link Robot Manipulator

Petar Kormushev[1], Yiannis Demiris[2], and Darwin G. Caldwell[3]

Abstract— **Encoders have been an inseparable part of robots since the very beginning of modern robotics in the 1950s. As a result, the foundations of robot control are built on the concepts of kinematics and dynamics of articulated rigid bodies, which rely on explicitly measuring the robot configuration in terms of joint angles – done by encoders.**

In this paper, we propose a radically new concept for controlling robots called Encoderless Robot Control (EnRoCo). The concept is based on our hypothesis that it is possible to control a robot without explicitly measuring its joint angles, by measuring instead the effects of the actuation on its end-effector. To prove the feasibility of this unconventional control approach, we propose a proof-of-concept control algorithm for encoderless position control of a robot's end-effector in task space. We demonstrate a prototype implementation of this controller in a dynamics simulation of a two-link robot manipulator. The prototype controller is able to successfully control the robot's end-effector to reach a reference position, as well as to track continuously a desired trajectory.

Notably, we demonstrate how this novel controller can cope with something that traditional control approaches fail to do: adapt on-the-fly to changes in the kinematics of the robot, such as changing the lengths of the links.

I. INTRODUCTION

Since the very beginning of modern robotics in the 1950s until present day, *encoders* have been an inseparable part of robots. Even the very first digitally operated robot 'Unimate', invented by George Devol in 1954, had encoders [1]. As a result, the foundations of modern robotics are built on the concepts of kinematics and dynamics of articulated rigid bodies, which rely on explicitly measuring the robot configuration in terms of joint angles – done by encoders.

In this paper, we propose a radically new concept for controlling robots called Encoderless Robot Control (EnRoCo). The concept is based on our hypothesis that it is possible to control a robot without explicitly measuring its joint angles, by measuring instead the effects of the actuation on its end-effector. This is a non-trivial and non-obvious statement. To make it clearer, a useful analogy from everyday life is a person driving a car - he does not need to explicitly measure the angle of the steering wheels in order to steer the car. Instead, he can infer it by observing the car's motion. Similarly, the proposed EnRoCo approach can control a robot by observing how the actuators affect the end-effector's motion, without explicitly measuring the joint angles.

Authors [1,3] are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), 16163 Genoa, Italy {petar.kormushev, darwin.caldwell}@iit.it.

Author [2] is with the Department of Electrical and Electronic Engineering, Imperial College London, United Kingdom y.demiris@imperial.ac.uk.

II. RELATED WORK

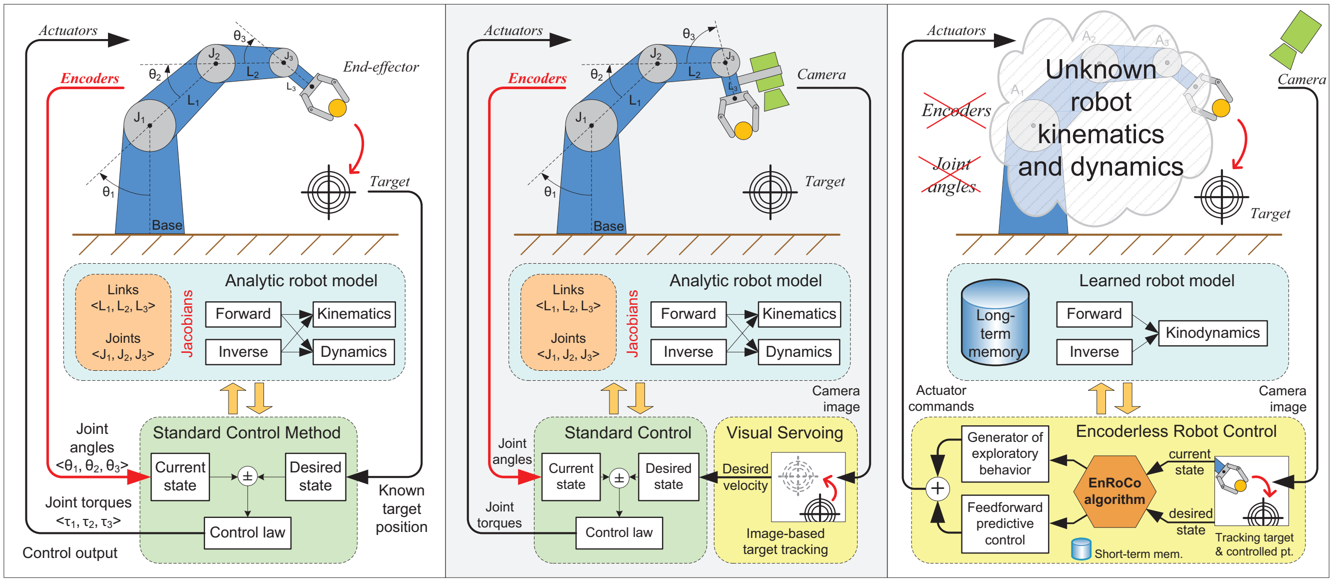
To the best of our knowledge, there is no existing *encoderless* robot control approach to this date that does not rely on any type of joint angle measurement or estimation. The reason for this stems from the well-established tradition in robotics and control theory, to try to model explicitly the system that needs to be controlled [2], [3]. For example, a very recent paper on the topic of encoderless robot motion control [4] still relies on joint angle estimation using the back electromotive force of motors.

Among the existing robot control methods, the one that is somehow closer to the proposed encoderless robot control is *visual servoing* [5]. However, the similarity between EnRoCo and visual servoing is only superficial, to the extent that both approaches use exteroception (external sensing - e.g. a camera) for observing the robot's motion. What they do with the exteroceptive information is very different. In a typical visual servoing control, the camera image is used for calculating a desired velocity for the end-effector, which is then sent to a conventional velocity controller that still uses the joint encoders to execute the motion.

Unlike existing methods, EnRoCo does not use encoders or joint angle estimation at all in the entire control architecture. Instead, EnRoCo uses an external camera (one or more if needed to deal with occlusions) to perceive the effects that the actuators have on the robot's motion, and then uses learning algorithms to decide what actuation signals need to be sent to the actuators in order to achieve the desired robot motion. Since EnRoCo does not need joint angle information, it does not make any assumptions regarding the kinematic structure of the robot, meaning that EnRoCo does not need *a priori* knowledge or model of the robot. This is the most important distinction between EnRoCo and existing control methods, as illustrated in Fig. 1.

In the field of industrial power electronics, there is an approach for sensorless/encoderless control of synchronous machines, such as electric motors [6]–[10]. The principle behind these approaches is dual-use of the motor simultaneously as an encoder. This is usually done by injecting high-frequency signal in the main control signal sent to the motor, and measuring the changes in the back-EMF. A similar dual-use principle for encoderless position measurement is based on hall effect sensor outputs of direct drive linear motors [11]. Another related approach is used for direct torque control of brushless reluctance machines [12]. These principles are completely different from the proposed encoderless robot control concept in this paper.

The main challenge in the robot control problem is the



(a) Conventional encoder-based feedback control

(b) Conventional visual servoing control

(c) Proposed encoderless robot control

Fig. 1. Illustration of the differences between the existing encoder-based robot control approaches - in (a) and (b), and the proposed encoderless robot control (EnRoCo) - in (c). Among the many differences, the most important one is that EnRoCo is the only approach that does not use encoder feedback (nor joint angle estimation) for controlling the robot. Instead, the feedback is done entirely through exteroception, and the human-designed analytic robot model is replaced by a self-learned model.

complexity of the dynamics and the numerous uncertainties. They arise from imprecise knowledge of kinematic and inertia parameters, from joint and link flexibility, actuator dynamics, friction, sensor noise, and so on [3]. Modelling explicitly all these effects by hand produces enormously complex models. Instead, EnRoCo is able to model all this autonomously and on-the-fly while the robot moves.

EnRoCo is not the first approach to use model learning for controlling a robot. For example, approaches like body-schema learning [13], learning forward models [14], motor babbling [15], and reinforcement learning of robot skills [16] employ machine learning techniques to help control a robot with unknown or uncertain kinematic/dynamic properties. However, unlike EnRoCo, all existing approaches ultimately rely on encoder (or joint angle) feedback for estimating the robot state (e.g. position, orientation, and velocity of the end-effector). Therefore, this is the first time an encoderless robot control concept is being proposed that does not use any joint angle estimation, to replace the conventional encoder-based feedback control architecture with a learning-based encoderless approach.

In the field of neuroscience, a very interesting study published in Nature [17] demonstrates how a monkey can learn to control a robot arm without using any encoder feedback. The robot is controlled by signals from the monkey's brain, and the only feedback available to the monkey is its own vision. Conceptually, what the monkey achieves by learning to control the robot arm purely based on visual feedback is very similar to the concept of the proposed EnRoCo controller. The fact that a monkey can learn to do this efficiently gives evidence in support of the feasibility of the proposed control concept.

III. PROPOSED APPROACH

A high-level conceptual flowchart of the proposed Encoderless Robot Control (EnRoCo) approach is shown in Fig. 2. The main idea is that it is possible to obtain information about the local combined kinematics and dynamics of the robot (what we call 'kinodynamics') by generating pseudo-random actuation control signals and observing their effect on the robot's end-effector. Then, after collecting sufficient observations, the local kinodynamics can be approximated and the EnRoCo controller can estimate what actuation control signal is required to make the end-effector move in a desired direction towards a given reference position. After each movement, the resulting effect on the end-effector's state is compared with the anticipated effect. If the difference is significant, this means that the local kinodynamics is not known precisely enough, which triggers a new exploratory phase. The most important components from Fig. 2 are as follows:

- 1 A decision is made whether to collect more information about the local kinodynamics (by triggering the generator of exploratory behavior) or to use the already collected information.
- 2 Based on the available local kinodynamics information, the EnRoCo controller is trying to predict what actuation signal would move the end-effector towards the given reference position. One possible way to calculate this is proposed in the next section.

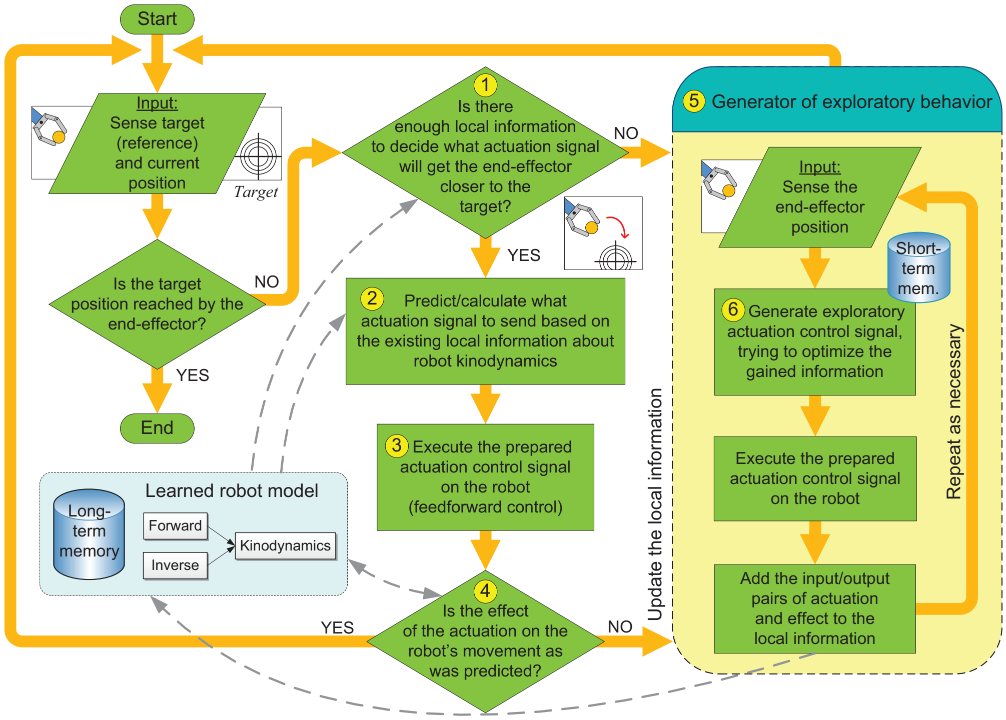


Fig. 2. A high-level diagram showing how the proposed Encoderless Robot Control (EnRoCo) approach works.

- (3) The calculated actuation signal from step (2) is executed on the robot. Please note that this is a feed-forward execution of the control signal (which could be torque, voltage, current, or other signal supported by the robot motor drivers) without any encoder/joint angle feedback.
- (4) The effect of the actuation from step (3) is compared to the predicted effect from step (2). If the prediction was not accurate enough, this triggers new exploratory phase which adds more local kinodynamics information which, in turn, improves the accuracy of the future predictions.
- (5) The generator of exploratory behavior works by generating pseudo-random actuation control signals which we call actuation primitives. These primitives have parameters (such as magnitude and duration) which can be modulated in order to produce different control signals.
- (6) The generator has its own short-term memory which helps to generate fewer primitives while simultaneously optimizing the gained information about the local kinodynamics. For example, one possibility is to generate primitives that are orthogonal in the space of primitive parameters.

IV. PROPOSED IMPLEMENTATION

The description of EnRoCo in Section III is rather abstract and it could be implemented in many different ways. In this section, we propose one concrete implementation of EnRoCo. To be more specific, we propose an EnRoCo implementation for a 2-degree-of-freedom serial robot manipulator.

The proposed implementation is based on *actuation primitives*. An actuation primitive produces a control signal $\tau(t)$ (could be actuation torque, voltage, current, etc.) that is sent to an actuator and is defined as a function of time:

$$\tau(t) = \begin{cases} \tau_p & \text{if } t \in [t_0, t_0 + \frac{d_p}{2}) \\ -\tau_p & \text{if } t \in [t_0 + \frac{d_p}{2}, t_0 + d_p] \\ 0 & \text{if } t \in (-\infty, t_0) \cup (t_0 + d_p, \infty) \end{cases} \tag{1}$$

where the parameter τ_p defines the magnitude (torque) of the actuation primitive, d_p defines the duration of the primitive, and t_0 denotes the starting time. Example primitives generated by EnRoCo are shown in Fig. 3. The proposed EnRoCo controller generates actuation primitives with different values for the parameters (τ_p and d_p) and sends them to each actuator of the robot. In the proposed implementation, the primitives are sent sequentially and synchronously for all actuators. In principle, it could also be possible to send them asynchronously and/or in an overlapping way.

Please note that the order of the actuators does not matter for the EnRoCo controller. In fact, the controller does not even know which actuator corresponds to which joint. By making no assumptions about the actuators and their relationship to the joints, the EnRoCo controller becomes agnostic to changes in the design of the robot, which is a significant advantage over traditional controller design.

While the EnRoCo controller is running, it is collecting a dataset $\{p_i\}$ of actuation primitives that have been executed on the robot, produced by the 'Generator of exploratory behavior' (step (5) in Fig. 2). Then, every time at step (2),

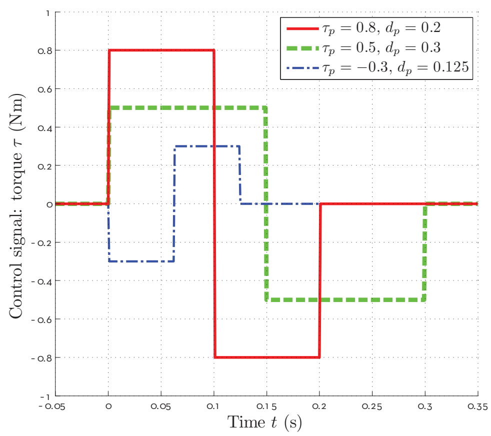


Fig. 3. Example actuation primitives as the ones used by the proposed Encoderless Robot Control implementation. Three primitives are shown, each with different parameters: duration d_p and magnitude τ_p. The starting time is fixed at $t_0 = 0$ for all of them for easier comparison.

the EnRoCo controller is estimating new parameters for an actuation primitive to execute next. Here we describe exactly how this is done.

Let $\hat{p}$ be the desired primitive whose parameters $\tau_p(\hat{p})$ we would like to estimate, in order to move the end-effector towards a desired goal position. The desired primitive includes two components - one for each actuator:

$$b_1 = \begin{bmatrix} \tau_p^1(\hat{p}) \\ \tau_p^2(\hat{p}) \end{bmatrix} \tag{2}$$

The idea is to represent $\hat{p}$ as a linear combination of the k-nearest neighbor (k-NN) primitives that have been previously executed and recorded in the long-term memory. Let $p_1 \dots p_k$ be these k-NN primitives. The distance is calculated from the current end-effector position to the starting position of each primitive $\{p_i\}$. Using some unknown weights $x = [x_0, x_1, \dots, x_k]^T$, the linear combination of the k-NN primitives can be expressed in matrix form as follows:

$$A_1 x = b_1, \tag{3}$$

where the matrix A_1 contains the parameters of the k-NN primitives:

$$A_1 = \begin{pmatrix} 1 & \tau_p^1(p_1) & \tau_p^1(p_2) & \cdots & \tau_p^1(p_k) \\ 1 & \tau_p^2(p_1) & \tau_p^2(p_2) & \cdots & \tau_p^2(p_k) \end{pmatrix}_{2\times(k+1)}, \tag{4}$$

where $\tau_p(p_i)$ is the magnitude of the i-th actuation primitive. In order to find suitable coefficients $\{x_i\}$ for the linear combination, we use the available information about the outcomes of the k-NN primitives:

$$A_2 = \begin{pmatrix} 1 & \Delta x(p_1) & \Delta x(p_2) & \cdots & \Delta x(p_k) \\ 1 & \Delta y(p_1) & \Delta y(p_2) & \cdots & \Delta y(p_k) \\ 1 & \Delta z(p_1) & \Delta z(p_2) & \cdots & \Delta z(p_k) \end{pmatrix}_{3\times(k+1)}, \tag{5}$$

where $[\Delta x(p_i)\ \Delta y(p_i)\ \Delta z(p_i)]^T$ is the relative displacement of the end-effector after the execution of the primitive p_i. Using the information about the current end-effector position and the target position, we can choose a specific desired effect for the primitive we are generating. For example, this effect can be either directly moving the end-effector to the final target position, or moving it towards the target at a certain distance. The selected effect is expressed in terms of the relative displacement of the end-effector as follows:

$$b_2 = \begin{bmatrix} \Delta x(\hat{p}) \\ \Delta y(\hat{p}) \\ \Delta z(\hat{p}) \end{bmatrix} \tag{6}$$

Next, we can obtain the coefficients $\{x_i\}$ by solving the following equation for x:

$$A_2 x = b_2 \tag{7}$$

Please note that this is not necessarily a well-posed problem, because the rank of matrix A_2 might not be full, and thus there might be many possible solutions for x. To go around this problem, we use least squares regression to solve it by finding the smallest (squared) vector x that is a solution. Then, the calculated value for x can be substituted in (3) and thus, finally, the desired primitive parameters $\tau_p(\hat{p})$ can be obtained from b_1.

The described mechanism is somewhat similar to the ARCHER algorithm [18], which also uses an iterative local regression process, and already works well. However, it does not take into account the different level of reliability (confidence) of the different k-NN primitives. For example, primitives that are very close[1] to the current state of the end-effector should be considered to be more reliable[2] than further away primitives. To take this important information into account, it is possible to apply instead a *weighted* least squares approach for solving (7). The idea is to put different weights $w_1 \dots w_k$ to the different neighbors $p_1 \dots p_k$, according to their distance[1] from the current state. This can be achieved by changing equations (3) and (7) to:

$$\begin{cases} A_1 W x = b_1 \\ A_2 W x = b_2 \end{cases} \tag{8}$$

where $W = \text{diag}(1, w_1, w_2, \dots, w_k)$. Please note that this differs from the usual weighted least squares approach in which the weights are put on feature columns, not on sample rows as in our case.

This concludes the description of the proposed EnRoCo implementation for encoderless position control of a two-link robot manipulator.

V. EXPERIMENTS IN SIMULATION

A dynamics simulation of a robot arm was performed using MATLAB/Simulink and a modified version of the

[1]The proximity between previously executed primitives and the current end-effector position is calculated using Euclidean distance between the beginning position of each primitive and the current end-effector position.

[2]The closest k-NN primitives are more reliable because they can better approximate the local robot kinodynamics than further away primitives, therefore they should be considered with higher weights when calculating the new primitive.

TABLE I

ROBOT KINEMATIC AND DYNAMIC SPECIFICATIONS

Link lengths	*Base*:	0.35 [m]	
	L_1:	1.00 [m]	1.20 [m]*
	L_2:	1.00 [m]	1.20 [m]*
Link masses	L_1:	1 [kg]	
	L_2:	1 [kg]	

* The kinematic changes for the experiment in Fig. 7.

Robotics Toolbox [19]. The kinematic and dynamic specifications of the simulated robot used for the experiments are shown in Table I. Please note that this information is not given to the EnRoCo controller in any way.

The EnRoCo controller was implemented as described in Section IV using the $k = 4$ nearest neighbor primitives and the weighted least squares method with weights $[1\ 0.75\ 0.5\ 0.25]^T$ for all experiments. Since the experiments were performed in simulation, there was no need to use a camera. Instead, the end-effector state was obtained internally from the simulator.

Three types of experiments were performed using the proposed EnRoCo controller implementation, as follows:

1) Reaching a single reference/target position with the end-effector (shown in Fig. 4 and Fig. 5);
2) Tracking a continuous reference trajectory with the end-effector (shown in Fig. 6);
3) Adapting to changes in the robot kinematics while tracking a reference trajectory (shown in Fig. 7).

The experiments and their results are explained in detail below. The video accompanying this paper contains motion sequences from all experiments. A longer version of the video is also available online [20].

A. Reaching a reference position

In these experiments, the EnRoCo controller is given a desired reference position that needs to be reached with the robot's end-effector. Four experiments of this type are shown in Fig. 4. In each experiment, the EnRoCo controller starts from a blank state (i.e. without any prior knowledge about the robot) and performs a brief exploratory behavior produced by the generator in step 5 of Fig. 2. Since the given reference position is very far from the current end-effector position, the EnRoCo controller generates intermediate reference/target positions that lie on a straight line between the end-effector and the ultimate target position. This is done at step 2 of Fig. 2. Each intermediate target is generated to be not too far from the current end-effector position, bounded above by a constant maximum distance (in these experiments 0.04 [m] was used). The reason for this is to limit the effect of each primitive to a relatively small neighborhood around the current end-effector position, because the robot kinodynamics is only known locally around this position to EnRoCo. As the EnRoCo controller explores bigger parts of the reachable state space of the robot, this maximum distance bound can be relaxed (i.e. increased). The specific value of this bound is not critical, because if the effect of a generated primitive differs substantially from the anticipated outcome, the EnRoCo controller will trigger automatically an exploratory behavior (step 4 of Fig. 2).

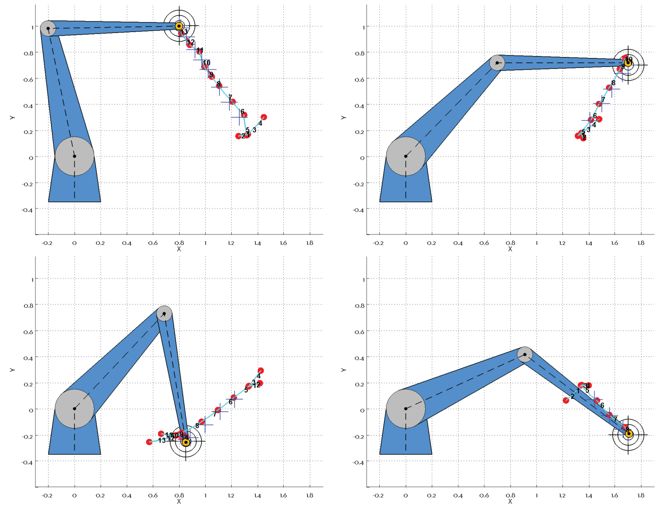


Fig. 4. These experiments demonstrate how the EnRoCo controller can reach a desired reference/target position for the end effector. Four experiments are shown, one for each of the four targets indicated by the big concentric circles with a cross. The numbers in the figure indicate the sequence of executed actuation primitives. The end-effector trajectory is indicated with a cyan line. The end-effector tip is shown with a yellow circle. The small blue crosses indicate the intermediate reference positions generated on a straight line towards the target position for each primitive. The actual positions reached by the end-effector at the end of each primitive are indicated with smaller red circles. In all four cases, the EnRoCo controller successfully reaches the target position. A video clip of these experiments is available online [20].

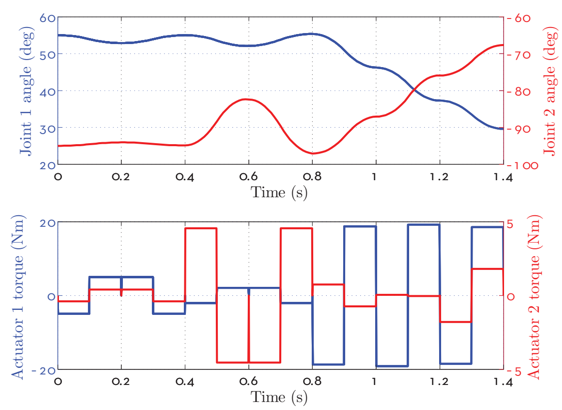


Fig. 5. The recorded joint positions (in degrees) and control signals (in Nm) sent to the actuators as a result of the generated actuation primitives. These data correspond to the last experiment in Fig. 4 in which the target position is in the bottom-right corner at coordinates (1.7, -0.2).

To demonstrate the generated actuation primitives, Fig. 5 shows the sequence of executed primitives and the recorded robot movement for the last of the four experiments from Fig. 4. In these experiments, the durations of the primitives are kept constant (at 0.2 [s]) and only their magnitude is adjusted by the EnRoCo controller.

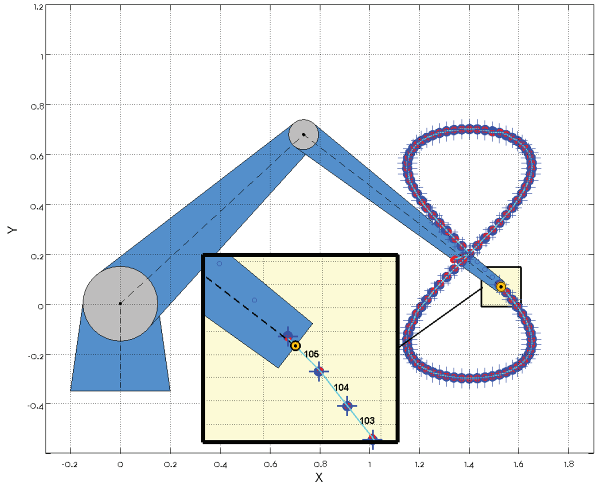


Fig. 6. This experiment shows how the EnRoCo controller can track a continuous reference trajectory. The reference figure-8 trajectory is indicated with blue circles. The end-effector trajectory is indicated with a cyan line. The end-effector tip is shown with a yellow circle. The zoomed-in square area shows a magnified view of the following: (i) three consecutive actuation primitives, numbered 103, 104 and 105; (ii) the three corresponding reference/target positions for each primitive, indicated with blue crosses; (iii) the actual positions reached by the end-effector, indicated with smaller red circles. A video clip of the experiment is available online [20].

B. *Tracking a continuous reference trajectory*

In these experiments, the EnRoCo controller is given a whole continuous reference trajectory that needs to be tracked by the robot's end-effector. Fig. 6 shows one experiment in which the trajectory has a figure-8 shape. The EnRoCo controller is able to track the reference trajectory in task space very well. Please note that the reference trajectory is specified only in the task space and not in the time domain. This allows the EnRoCo controller to take its time to perform exploratory behavior as necessary, without penalizing the tracking performance. The goal of this experiment is also to demonstrate that EnRoCo can reuse the accumulated knowledge about the robot kinodynamics which is reflected by the reduced need for exploratory behavior when the robot re-visits previously explored areas of the state space.

C. *Adapting to changes in the robot kinematics*

One of the most important advantages of EnRoCo over conventional encoder-based controllers is its non-reliance on prior kinematics knowledge. This means that EnRoCo does not need robot kinematics information and does not use conventional forward/inverse kinematic calculation in order to control the robot's motion. To demonstrate this, we conducted a trajectory-tracking experiment during which the kinematics of the robot is changed. In particular, the lengths of two links are increased by 20%, as indicated in Table I. A conventional inverse-kinematics-based controller cannot cope with such a change on-the-fly, because its internal kinematics model needs to be manually updated to reflect the changed kinematics. EnRoCo, on the other hand, automatically adapts to the changes and continues tracking the trajectory after a brief exploratory period. The comparison between the conventional controller and EnRoCo is shown in Fig. 7.

VI. DISCUSSION

The proposed proof-of-concept implementation of EnRoCo clearly demonstrates the feasibility of the proposed novel robot control approach. However, the proposed implementation has a few drawbacks. For example, it does not take into account the effects of gravity. Because of this, the motion of the two-link manipulator was kept in a horizontal plane during the presented experiments. Another limitation is the fact that the actuation primitives are executed synchronously and are kept with a constant duration. In the future, we plan to investigate more elaborate algorithms that are able to modify not only the magnitude, but also the duration of the actuation primitives.

The goal of the proposed encoderless robot control approach is not necessarily to eliminate the use of encoders, but to design a robot controller that can work even without encoders. A straightforward application would be as a fail-safe controller.

Currently, robots are designed with stiff links to avoid bending (in order for the kinematic calculations to work). By not relying on encoders for controlling robots, EnRoCo will open up exciting possibilities for the mechanical design of future robots. For example, the links will no longer need to be so stiff, and the kinematics will no longer need to be fixed. As an illustration, imagine a lightweight prosthetic arm or a robot exoskeleton that can grow, bend, and adapt to accommodate its patient. Such a device would be very difficult to control with the existing control methods.

Moreover, EnRoCo can provide robot cost reduction by eliminating the need for encoders thus simplifying their design. Further applications include robotic micro-manipulation, where a major difficulty is the miniaturization of encoders, and EnRoCo could eliminate them altogether.

To summarize, the benefits from breaking the dependency on joint encoders are surprisingly many, not only for controlling existing robots, but also for the design of new robots. Potential applications include lower-cost robots due to simpler design, safer human-robot interaction due to lighter robots, modular and reconfigurable robots whose kinematics changes over time (e.g. evolving hardware), etc.

VII. CONCLUSION

We have presented a novel approach for encoderless control of a robot manipulator. The approach does not rely on any joint angle information or estimation and does not require any *a priori* knowledge about the robot kinematics or dynamics. The approach works by generating actuation primitives and perceiving their effect on the robot's end-effector, thereby building a local kinodynamic model of the robot. We have proposed a concrete implementation of this approach and conducted proof-of-concept experiments with it in simulation. The experimental results show that this novel control approach can reach a reference position and track a continuous reference trajectory. More importantly, the

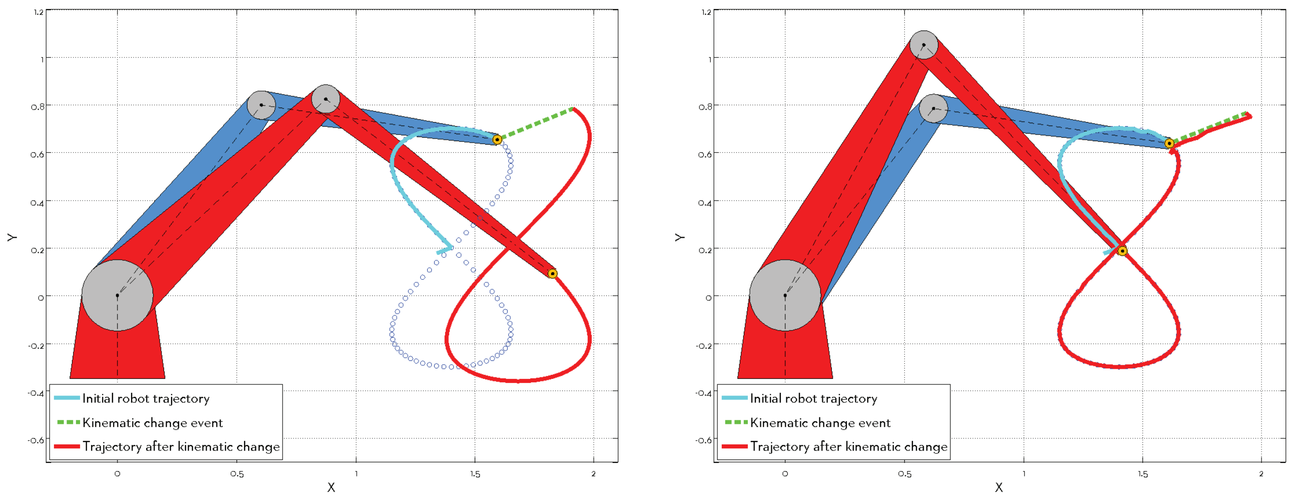


Fig. 7. This experiment shows how the EnRoCo controller copes with changes in the robot kinematics on-the-fly, while tracking a reference trajectory. The reference figure-8 trajectory is indicated with blue circles. The time and place where the kinematics change event occurs is indicated with the green dashed line. *Left:* A conventional inverse-kinematics-based position controller cannot adapt to the changes in the kinematics and fails to track the reference trajectory after the kinematics change event. *Right:* The proposed EnRoCo controller automatically adapts to the changes in the kinematics and successfully tracks the reference trajectory even after the kinematics change event. In both cases, the *blue robot* is the original robot that starts the movement, and the *red robot* is the kinematically-modified robot which has 20% longer lenghts of links L_1 and L_2, compared to the initial (blue) robot. Neither one of the two controllers is notified about the kinematics change. The end-effector trajectory is shown with a cyan line before, and with a red line after the kinematics change event. A video clip of the experiment is available online [20].

controller can adapt on-the-fly to changes in the robot kinematics, which is something very difficult for conventional controllers. The proposed control approach looks promising and has many potential applications not only for the control of existing robots, but also for new robot designs.

Regarding future work, a major issue remaining to be investigated is scalability of EnRoCo to multi-DOF robot manipulators, as well as practical real-world implementation issues, both of which are beyond the scope of this introductory paper.

References

[1] G. C. Devol, "Programmed article transfer," June 1961, US Patent 2,988,237.

[2] B. Siciliano, L. Sciavicco, and L. Villani, *Robotics: modelling, planning and control*. Springer Verlag, 2009.

[3] W. S. Levine, Ed., *The Control Handbook (Second Edition)*. CRC Press, Inc., 2010.

[4] A. Kawamura, M. Tachibana, S. Yamate, and S. Kawamura, "Encoderless robot motion control using vision sensor and back electromotive force," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS 2014)*, Chicago, USA, September 2014.

[5] F. Chaumette, S. Hutchinson, *et al.*, "Visual servo control, Part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[6] O. C. Ferreira and R. Kennel, "Encoderless control of industrial servo drives," in *Power Electronics and Motion Control Conference, 2006. EPE-PEMC 2006. 12th International*, Aug 2006, pp. 1962–1967.

[7] R. Morales-Caporal and M. Pacas, "Encoderless predictive direct torque control for synchronous reluctance machines at very low and zero speed," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 12, pp. 4408–4416, 2008.

[8] S. Murakami, T. Shiota, M. Ohto, K. Ide, and M. Hisatsune, "Encoderless servo drive with adequately designed ipmsm for pulse-voltage-injection-based position detection," *Industry Applications, IEEE Transactions on*, vol. 48, no. 6, pp. 1922–1930, 2012.

[9] P. Landsmann, C. Hackl, and R. Kennel, "Eliminating all machine parameters in encoderless predictive torque control without signal injection," in *Electric Machines & Drives Conference (IEMDC), 2011 IEEE International*. IEEE, 2011, pp. 1259–1264.

[10] J. Holtz, "Sensorless control of induction machines With or without signal injection?" *Industrial Electronics, IEEE Transactions on*, vol. 53, no. 1, pp. 7–30, 2006.

[11] E. A. Baran, T. E. Kurt, and A. Sabanovic, "Lightweight design and encoderless control of a miniature direct drive linear delta robot," in *Electrical and Electronics Engineering (ELECO), 2013 8th International Conference on*. IEEE, 2013, pp. 502–506.

[12] M. G. Jovanovic, J. Yu, and E. Levi, "Encoderless direct torque controller for limited speed range applications of brushless doubly fed reluctance motors," *Industry Applications, IEEE Transactions on*, vol. 42, no. 3, pp. 712–722, May 2006.

[13] M. Hoffmann, H. G. Marques, A. Hernandez Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, "Body schema in robotics: a review," *Autonomous Mental Development, IEEE Transactions on*, vol. 2, no. 4, pp. 304–324, 2010.

[14] A. Dearden and Y. Demiris, "Learning forward models for robots," in *IJCAI*, vol. 5, 2005, p. 1440.

[15] Y. Demiris and A. Dearden, "From motor babbling to hierarchical learning by imitation: a robot developmental pathway," 2005.

[16] P. Kormushev, S. Calinon, and D. G. Caldwell, "Reinforcement learning in robotics: Applications and real-world challenges," *Robotics*, vol. 2, no. 3, pp. 122–148, 2013.

[17] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–1101, 2008.

[18] P. Kormushev, S. Calinon, R. Saegusa, and G. Metta, "Learning the skill of archery by a humanoid robot iCub," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, Nashville, TN, USA, December 2010, pp. 417–423.

[19] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011, ISBN 978-3-642-20143-1.

[20] Video accompanying this paper. [Online]. Available: http://kormushev.com/goto/ICRA-2015-Encoderless/