Kinematic-free Position Control of a 2-DOF Planar Robot Arm

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

***Abstract*— This paper challenges the well-established assumption in robotics that in order to control a robot it is necessary to know its kinematic information, that is, the arrangement of links and joints, the link dimensions and the joint positions. We propose a kinematic-free robot control concept that does not require any prior kinematic knowledge. 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.**

We implement a proof-of-concept encoderless robot controller and apply it for the position control of a physical 2-DOF planar robot arm. The prototype controller is able to successfully control the robot to reach a reference position, as well as to track a continuous reference trajectory. Notably, we demonstrate how this novel controller can cope with something that traditional control approaches fail to do: adapt to drastic kinematic changes such as 100% elongation of a link, 35-degree angular offset of a joint, and even a complete overhaul of the kinematics involving the addition of new joints and links.

I. INTRODUCTION

The foundations of modern robotics are built on the concepts of kinematics and dynamics of articulated rigid bodies. Practically every robotics textbook starts with a description of robot configuration using joint angles, and then uses them to introduce robot kinematics, dynamics, and control [1].

A major consequence of this is the implicit assumption that in order to control a robot, it is necessary to know its kinematic information, that is, the arrangement of links and joints, the link dimensions and the joint positions. Assuming, in addition, that the link dimensions are constant, then the only information needed to control a robot is the joint angles.

For more than 60 years this has been the accepted norm in robotics. In fact, even the very first digitally operated robot 'Unimate', invented by George Devol in 1954, already had an internal kinematics model and physical encoders to measure its joint angles for control purpose [2]. Now, 60 years later, robot controllers still rely on the same principle - of known kinematics and measurable joint positions.

In this paper, we challenge this well-established tradition in robotics. We demonstrate empirically that it is possible to control a robot without knowing any prior kinematic information and without measuring its joint angles. This is what we call *'kinematic-free robot control'*.

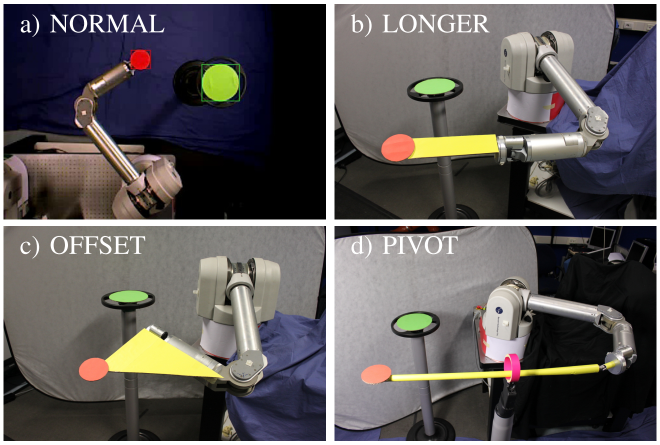


Fig. 1. Experimental setup for testing the proposed kinematic-free position controller. The goal of the proposed controller is to move the controlled point (marked with a red circle) to the target position (marked with a green circle). There is a fixed camera mounted above the robot looking down (not shown here). A sample image from the camera is given in (a). Four sets of experiments are conducted using four different kinematic configurations: (a) 'NORMAL' configuration where the controlled point coincides with the original end-effector of the robot; (b) 'LONGER' configuration where the second link is elongated by 100%; (c) 'OFFSET' configuration where a 35-degree offset is added to the second joint; (d) 'PIVOT' configuration where an additional link, a universal joint and a pivoting joint have been added to the robot.

II. KINEMATIC-FREE ROBOT CONTROL

In our preliminary theoretic work [3], we hypothesized the possibility of controlling a robot without explicitly measuring its joint angles. In this paper, for the first time, we present empirical evidence from real-world experiments that this hypothesis is true. We implement a kinematic-free robot controller that is able to control the position of a two-degree-of-freedom planar robot arm (Fig. 1).

The proposed kinematic-free position controller does not need any prior kinematic information about the robot, apart from the number of degrees of freedom (two). Moreover, the controller works without explicitly measuring the robot's joint angles. Since it does not rely on a fixed kinematics model, this kinematic-free position controller is robust to changes in the robot kinematics. For example, we demonstrate that it can cope with drastic kinematic changes such as 100% elongation of a link, 35-degree angular offset of a joint, and even a complete overhaul of the kinematics involving the addition of new joints and links (Fig. 1 and 2).

Instead of measuring the robot's joint positions, the controller measures instead the effects of the actuation on its end-effector. It applies exploratory control inputs (torques) to the actuators and observes the resulting end-effector motion

[1] Dyson School of Design Engineering, Imperial College London, SW7 2AZ, United Kingdom. E-mail: p.kormushev@imperial.ac.uk

[2] Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, United Kingdom

[3] Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genoa, Italy

using an external camera. This way it learns on-the-fly the robot's combined kinematics and dynamics, and is able to estimate what control input is needed in order to move the end-effector to a desired position.

To make this working principle clearer, a useful analogy from everyday life is a person driving a car. The driver does not need to explicitly measure the angle of the steering wheels in order to steer the car. Instead, he/she can infer it by observing the car's motion. Similarly, the proposed robot controller works by observing how the actuators affect the end-effector's motion and is able to control it without explicitly measuring the joint angles. Since the robot encoders are not being used[1], we call this approach *Encoderless Robot Control* (EnRoCo).

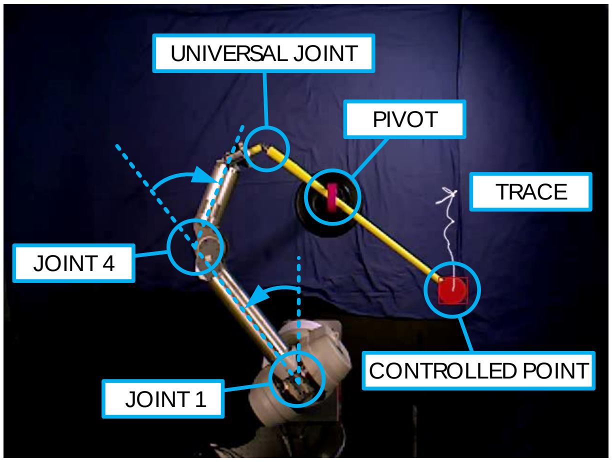


Fig. 2. Details about the PIVOT configuration shown in Fig. 1d. The Barrett WAM robot is modified by adding a new universal joint attached to the original end-effector, which is then connected to a new link (in yellow). The link goes through a sliding and pivoting joint (in magenta). The controlled point is moved to the end of the new link. Both new joints are passive, which means that the total number of actuated DOF remains the same (two). As the pivot point's position is fixed, the total number of DOF of the controlled point remains unchanged (two), but the kinematics and dynamics of the robot is drastically changed.

III. RELATED WORK

Given the full range of robot types in existence, in this paper we focus on an important subset of them - serial robot manipulators. By far, this is the most common type of robot today, and is the main building block of other robots (e.g. humanoids have limbs that are, essentially, manipulators).

To the best of our knowledge, there is no existing *kinematic-free* or *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 [1], [4]. For example, a very recent paper on the topic of encoderless robot motion control [5] 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* [6]. 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 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. 3.

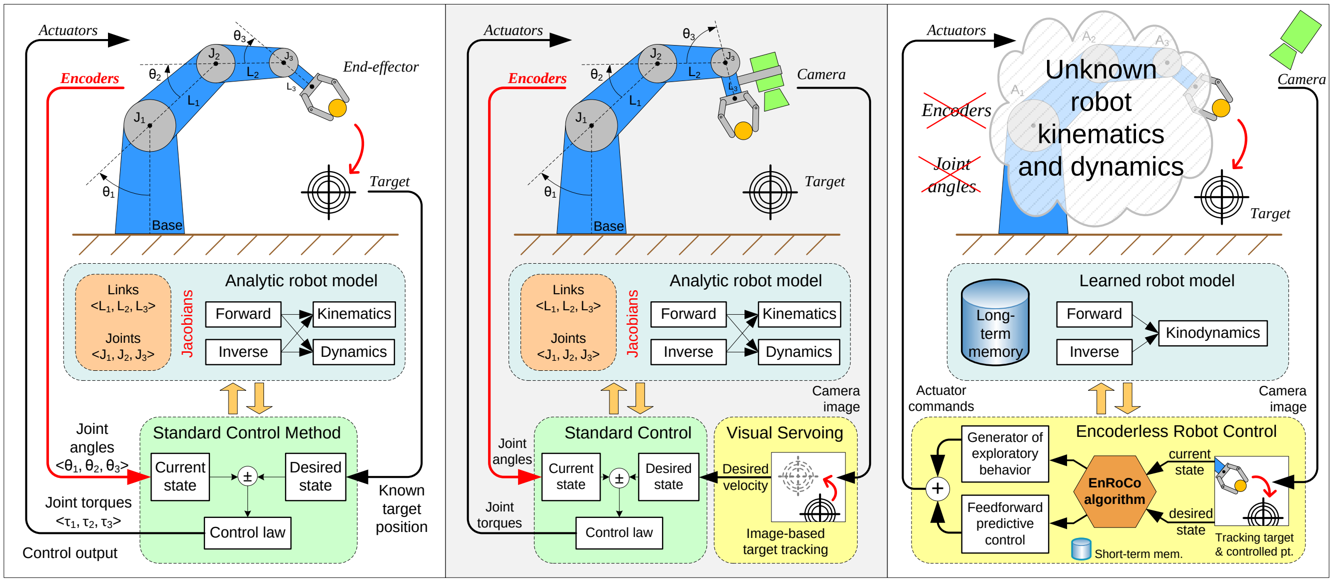
In the field of industrial power electronics, there is an approach for sensorless/encoderless control of synchronous machines, such as electric motors [7]–[11]. The principle behind these approaches is dual-use of the motor simultaneously as an encoder. This is usually done by injecting high-frequency 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 [12]. Another related approach is used for direct torque control of brushless reluctance machines [13]. These principles are completely different from the proposed encoderless robot control concept in this paper.

EnRoCo is not the first approach to use model learning for controlling a robot. For example, approaches like body-schema learning [14] and learning forward models [15] 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 implemented that does not use any joint angle estimation, to replace the conventional encoder-based feedback control architecture with a learning-based encoderless approach.

IV. PROPOSED APPROACH

A high-level conceptual flowchart of the proposed Encoderless Robot Control (EnRoCo) approach is shown in Fig. 4. The main idea is that it is possible to obtain information about the local combined kinematics and

[1] Although, physically, the Barrett WAM robot we use for conducting the experiments has built-in encoders, we do not use them in the implementation of our controller for any of the presented experiments.



(a) Conventional encoder-based feedback control

(b) Conventional visual servoing control

(c) Proposed encoderless robot control

Fig. 3. 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.

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. 4 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.
- ③ The calculated actuation signal from step ② is executed on the robot. Please note that this is a feedforward 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.

V. PROPOSED IMPLEMENTATION

The description of EnRoCo in Section IV 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 the 2-DOF modified Barrett WAM robot shown in Fig. 1.

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:

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

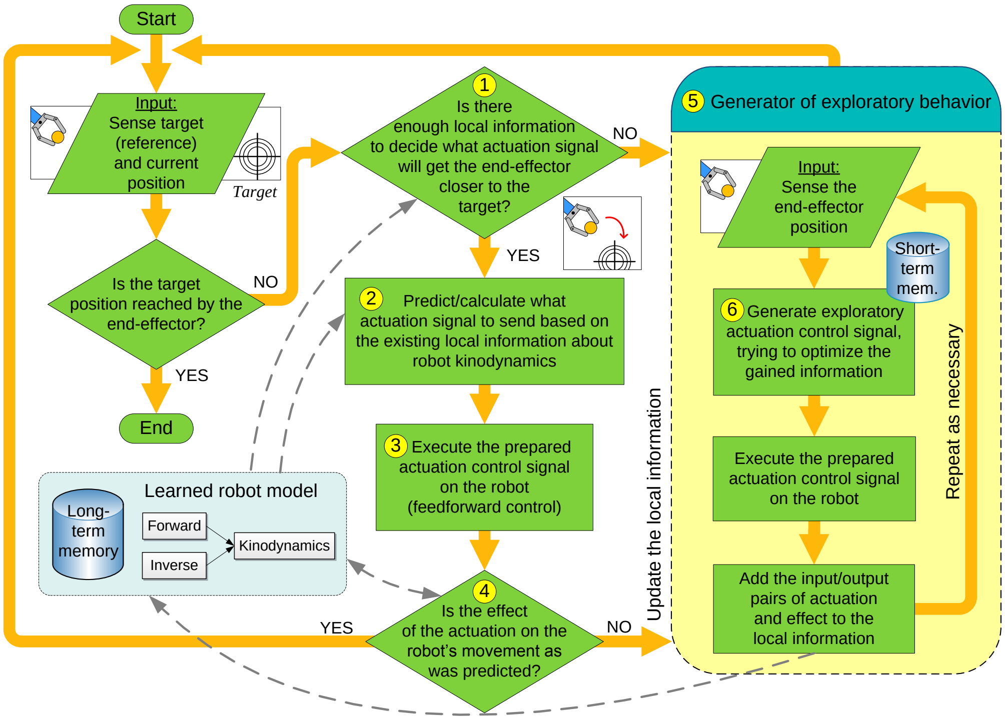


Fig. 4. A high-level diagram showing how the proposed Encoderless Robot Control (EnRoCo) approach works.

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. 5. 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.

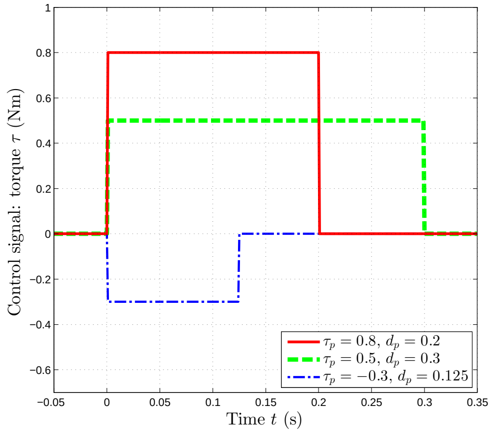


Fig. 5. Example actuation primitives as the ones used by the proposed Encoderless Robot Control implementation. Three primitives are shown, each with different parameters: duration d_p and magnitude τ_p. The starting time is fixed at $t_0 = 0$ for all of them for easier comparison.

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. 4). Then, every time at step ②, 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.

VI. EXPERIMENTS

The experimental setup for testing the proposed encoderless position controller is shown in Fig. 1. The robot is a modified Barrett WAM 7-DOF robotic arm that has had all degrees of freedom locked in a fixed position except two joints (joint 1 - base, and joint 4 - elbow), so that the remaining robot motion is restricted to a horizontal plane. This removes the effect of gravity and effectively turns the robot into a 2-DOF planar robot arm.

A fixed RGB-D camera (Asus Xtion Pro Live) is mounted above the robot looking down. The camera provides color images with 640x480 pixels resolution at 30 Hz. A sample image from the camera is shown in Fig. 1a. The point that needs to be controlled is marked with a red circle.

Two types of experiment are conducted: (i) reaching a desired position, and (ii) tracking a desired reference trajectory. For the first type, the target position is specified by placing a green circle on it. For the second type, the reference trajectory is programmatically specified in advance, according to the size of the accessible to the robot workspace. Please note that the reference trajectory is specified only in the task space and not in the time domain.

The position of the controlled point (red circle) and the target position (green circle) are tracked by color-based visual blob tracker. We use a ROS implementation of CMVision [16] that publishes the detected markers on a ROS topic.

For the implementation of the encoderless position controller we use MATLAB. The controller is implemented as described in Section V using the $k = 4$ nearest neighbor primitives. All experiments are conducted with the exact same settings of the controller. The communication between the controller, the visual blob tracker, and the robot is done through ROS topics. Due to the latency in processing and communication (mainly on the MATLAB side), the control frequency is reduced to only about 1 Hz. Despite this extremely low control frequency, the proposed controller manages to perform the planned tasks, as described below.

Four sets of experiments are conducted using four different kinematic configurations, as shown in Fig. 1. For convenience, we code-name the four sets as: NORMAL, LONGER, OFFSET and PIVOT. Below we give details for each of the four sets. The video accompanying this paper contains motion sequences from all experiments. A longer version of the video is available online [17].

A. Experiments with NORMAL configuration

In the NORMAL configuration the controlled point coincides with the original end-effector of the robot, as indicated by the red circle in Fig. 1a. The reachable workspace of this kinematic configuration is demonstrated in Fig. 6a, where a person holding the arm has swept the available workspace. This is only for illustration purpose and is not needed by the controller.

Starting from a *tabula rasa* state, the EnRoCo controller is first tested on 10 tasks for reaching a desired position. Before each test, the target position (green marker) is moved randomly to a new position inside the reachable workspace. Fig. 6b and 6c show two such individual tasks. Each individual task starts with an exploratory phase, which for all experiments has been fixed to two exploratory moves and their reverse moves[2]. Fig. 6d shows a consecutive run of 5 tasks, superimposed, after the end of the initial testing.

In all tests the EnRoCo controller manages to move the controlled point to the desired target position. The number of exploratory phases differs from task to task, depending on the distance between the starting point and the target position, as well as on the previous experience of the controller in this particular part of the workspace. Over time, as the experience of the controller grows, the need for exploration is naturally reduced. This is automatically done by the EnRoCo controller without any human intervention. For example, no exploration was needed to complete the consecutive tasks in Fig. 6d, since at that time the controller had accumulated enough information about the kinodynamics of the controlled point.

Next, an experiment about tracking a continuous reference trajectory is conducted. The reference trajectory is manually designed in the shape of figure-8, in order to cover as big part

[2] By 'reverse move' we mean applying the opposite actuation primitive of the last one by inverting the signs of all torque values.

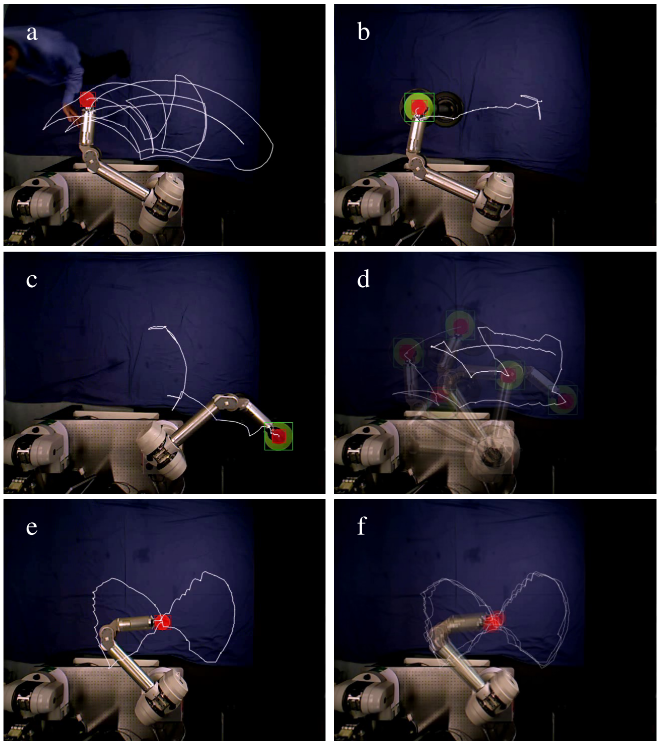


Fig. 6. Experiments conducted with the NORMAL configuration. The trace of the red marker is indicated in white. The EnRoCo controller demonstrates excellent target-reaching and trajectory-tracking abilities. A video clip of these experiments is available online [17].

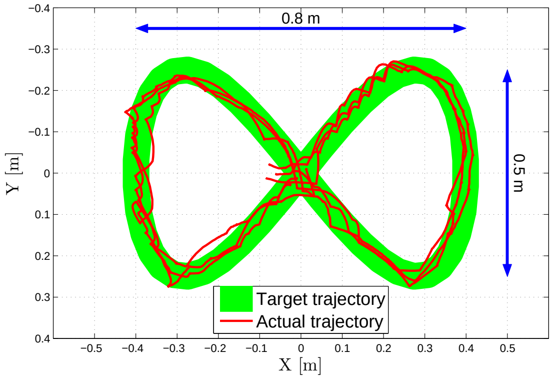


Fig. 7. The recorded trajectories of the controlled point (in red) in NORMAL configuration plotted against the reference figure-8 trajectory (in green). These data correspond to the experiment in Fig. 6f.

of the workspace as possible. A single attempt to track this trajectory is shown in Fig. 6e. Although not entirely smooth, the tracking is extremely good bearing in mind the low control frequency (1 Hz) and the complete lack of kinematic information. Furthermore, the superposition of multiple runs of the same tracking task shown in Fig. 6f reveals a very low variance among trials. The trajectories are shown in more detail in Fig. 7. The data show that EnRoCo exhibits high repeatability, again considering the poor quality of its available resources.

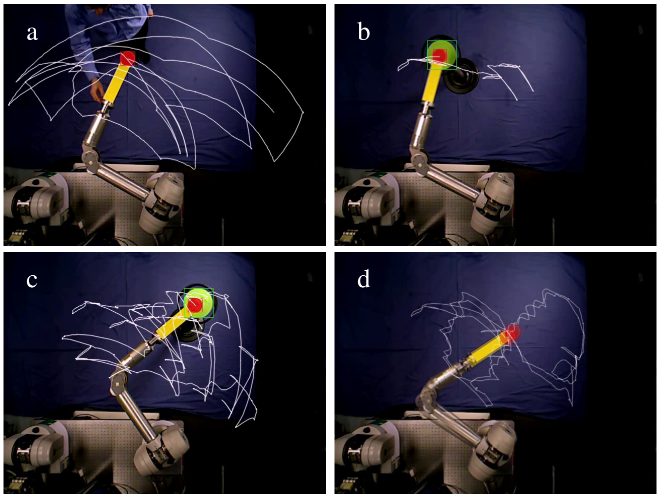


Fig. 8. Experiments conducted with the LONGER configuration. The workspace is bigger compared to the NORMAL configuration. The EnRoCo controller demonstrates satisfactory target-reaching and trajectory-tracking abilities, but worse than in NORMAL configuration.

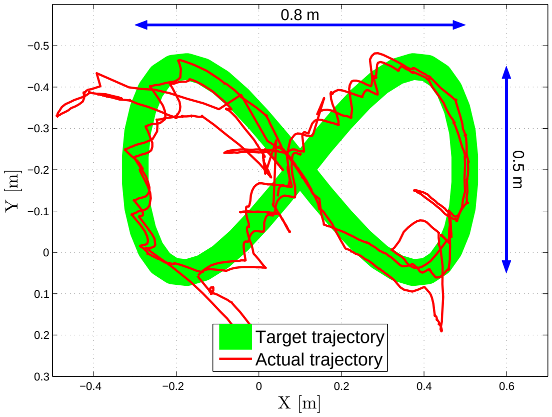


Fig. 9. The recorded trajectories of the controlled point (in red) in LONGER configuration plotted against the reference figure-8 trajectory (in green). These data correspond to the experiment in Fig. 8d.

B. *Experiments with LONGER configuration*

In the LONGER configuration the second link of the robot arm is elongated by 100%, as indicated by the yellow rectangle in Fig. 1b. The reachable workspace of this kinematic configuration is demonstrated in Fig. 8a, and is much bigger than the workspace in the NORMAL configuration (Fig. 6a).

Fig. 8b shows an individual task for target reaching, starting from a *tabula rasa* state of the controller. Fig. 8c shows a consecutive run of 15 target-reaching tasks, superimposed. In all tests the EnRoCo controller manages to move the controlled point to the desired target position. However, it is evident that the twice longer link increases the variance of the controlled point position by amplifying the actuation noise. This is more obvious in Fig. 8d which shows a superposition of multiple runs of the figure-8 trajectory tracking task. The same trajectories are shown in more detail in Fig. 9. The conclusion is that in the LONGER configuration the EnRoCo controller exhibits higher noise and variability with respect

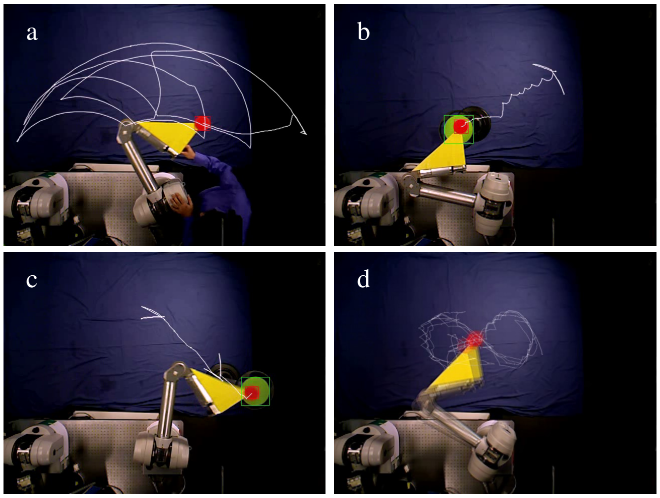


Fig. 10. Experiments conducted with the OFFSET configuration.

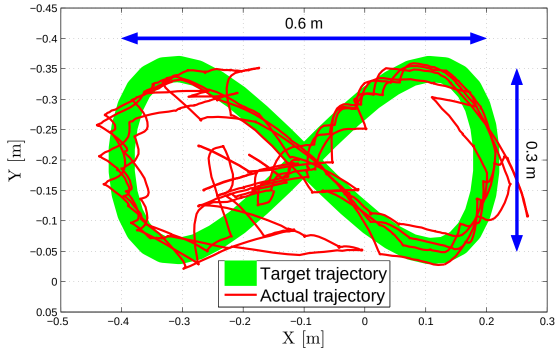


Fig. 11. The recorded trajectories of the controlled point (in red) in OFFSET configuration plotted against the reference figure-8 trajectory (in green). These data correspond to the experiment in Fig. 10d.

to the NORMAL configuration, but nevertheless it succeeds in reaching the desired targets.

C. Experiments with OFFSET configuration

In the OFFSET configuration a 35-degree offset is added to the second joint of the robot, as indicated by the yellow triangle in Fig. 1c. It is not quite intuitive, but the reachable workspace of this kinematic configuration is actually reduced compared to the NORMAL configuration. It is shown in Fig. 10a. The reason is the fact that the arm cannot bend past the self-collision point, which prevents the controlled point from reaching positions close to the base of the robot.

Fig. 10b and 10c show two individual tasks for target reaching, starting from a *tabula rasa* state of the controller. It is worth noting that the EnRoCo controller does not need more exploratory phases than in the previous configurations, which once again confirms the fact that this controller is agnostic to the kinematics of the robot and is not affected by joint angle offsets.

In all tests the EnRoCo controller manages to move the controlled point to the desired target position. Fig. 10d shows a superposition of multiple runs of the figure-8 trajectory tracking task in this robot configuration. The same trajectories are shown in more detail in Fig. 11. It exhibits higher variance compared to the NORMAL configuration, but is comparable to the LONGER configuration.

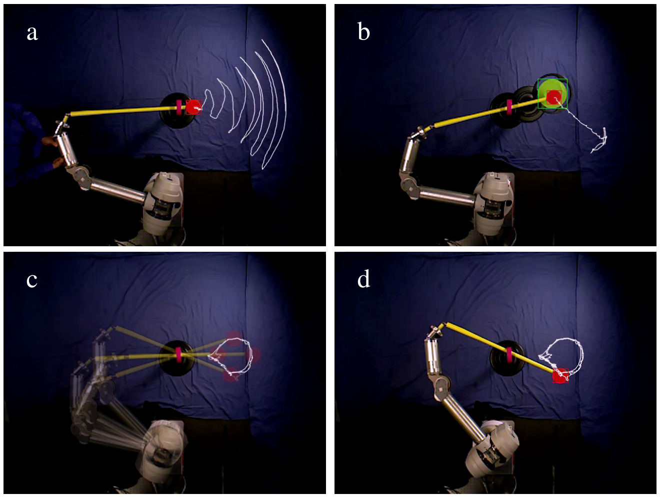


Fig. 12. Experiments conducted with the PIVOT configuration.

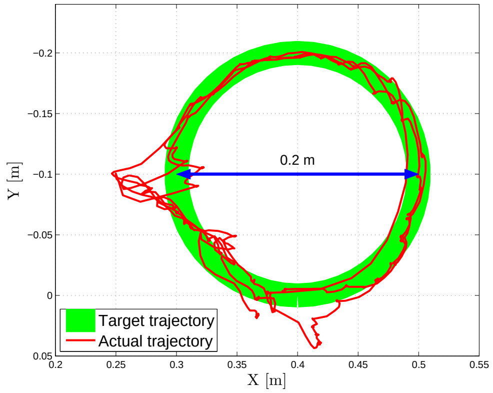


Fig. 13. The recorded trajectories of the controlled point (in red) in PIVOT configuration plotted against the reference circular trajectory (in green). These data correspond to the experiment in Fig. 12d.

D. Experiments with PIVOT configuration

In the PIVOT configuration an additional link, a universal joint and a pivoting joint have been added to the robot. Fig. 2 shows details about the mechanical overhaul of the robot arm. This is a major kinematic change and it really puts the proposed controller to a challenging test.

This time, the reachable workspace has a completely different shape, as shown in Fig. 12a. What cannot be easily seen from this static snapshot is that the velocity profile of the controlled point varies substantially throughout the workspace. This is due to the pivot point which, depending on the lever arm length, either increases or reduces the controlled point velocity with respect to the original end-effector velocity on the opposite side of the pivot point.

Despite this very challenging task, the EnRoCo controller manages flawlessly to execute both types of tasks: target-reaching tasks (an example given in Fig. 12b), and trajectory-following task (an example given in Fig. 12c). Due to the

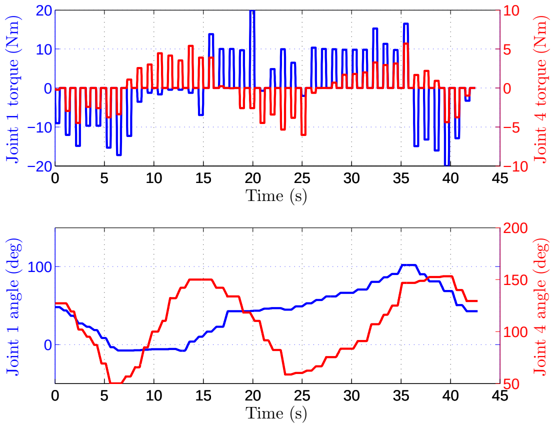


Fig. 14. *Top:* The recorded control signals (joint torques, in Nm) sent to the actuators as a result of the generated actuation primitives. *Bottom:* The respective joint positions (in degrees) of the two actuated DOFs. These data correspond to the trajectory-tracking experiment in Fig. 6e.

limited workspace, the figure-8 reference trajectory has been replaced with a circular trajectory for this configuration.

This experiment really brings out the advantages of the proposed controller with respect to conventional control approaches. Moreover, Fig. 13 shows that the controller achieves excellent tracking accuracy for such a small length scale of the reference trajectory (only 10 cm circle radius).

To demonstrate the generated actuation primitives, Fig. 14 shows the sequence of executed primitives and the recorded robot movement for the experiment in Fig. 6e. In all experiments, the durations of the primitives are kept constant and only their magnitude is adjusted by the EnRoCo controller.

VII. DISCUSSION

The proposed proof-of-concept implementation of EnRoCo clearly demonstrates the feasibility of the proposed novel kinematic-free robot control approach. However, the proposed implementation has a number of drawbacks. For example, it does not take into account the effects of gravity. 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.

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. Other 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), and fail-safe controllers that can work even without encoders.

VIII. CONCLUSION

We have presented a novel concept for kinematic-free control of a robot arm. The implemented encoderless robot controller 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. The experimental results with this proof-of-concept controller show that it can successfully control the position of the robot. More importantly, it can adapt even to drastic 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.

REFERENCES

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

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

[3] P. Kormushev, Y. Demiris, and D. G. Caldwell, "Encoderless position control of a two-link robot manipulator," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Seattle, USA, 2015.

[4] W. S. Levine, Ed., *The Control Handbook*. CRC Press, Inc., 2010.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] J. Bruce, T. Balch, and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots," in *Proc. IEEE/RSJ Intelligent Robots and Systems (IROS 2000)*, vol. 3, 2000, pp. 2061–2066.

[17] Video accompanying this paper. [Online]. Available: http://kormushev.com/goto/IROS-2015-Kinematic-free/