

# Model Predictive Control of a Magnetically Guided Rolling Microrobot

**Journal Article****Author(s):**

Pieters, Roel S.; Lombriser, Sievi; Alvarez-Aguirre, Alejandro; Nelson, Bradley J.

**Publication date:**

2016-01

**Permanent link:**

<https://doi.org/10.3929/ethz-b-000113412>

**Rights / license:**

[In Copyright - Non-Commercial Use Permitted](#)

**Originally published in:**

IEEE Robotics and Automation Letters 1(1), <https://doi.org/10.1109/LRA.2016.2521407>

**Funding acknowledgement:**

268004 - Microrobotics and Nanomedicine (EC)

# Model Predictive Control of a Magnetically Guided Rolling Microrobot

Roel Pieters, Sievi Lombriser, Alejandro Alvarez-Aguirre, and  
Bradley J. Nelson

**Abstract**—In this work, we present a model predictive control (MPC) approach to the motion control of a nonholonomic rolling microrobot. The mobile microrobot is designed to follow a rotating magnetic field such that it rolls on a supporting surface in a low Reynold’s number liquid environment. The MPC framework takes the nonholonomic and kinematic constraints of the microrobot and the actuation limitations of the system into account for the task of trajectory tracking and obstacle avoidance. We verify our approach by presenting a simulation study and experimental results with an 8-coil electromagnetic actuation system.

## I. INTRODUCTION

Manipulation and control at the microscale is characterized by the microrobot’s local surrounding environment and the constraints that apply due to this. Because of the difference in scale compared to macroscale robotics, surface effects have a bigger influence when objects scale down and when submerged in a fluid viscous forces dominate over inertial forces [1]. The advantage that microscale robotics has over macroscale is the relatively minor force necessary for actuation, and the capability of non-contact actuation principles. A general separation of actuation techniques for microrobotics divides approaches in either electrostatic [2], electromagnetic [3] or optical [4] techniques, among others. Despite this difference in actuation techniques the general problem of motion control for microscale robotics involves effects negligible in macroscale robotics. Surface and environmental effects and limitations in actuation disturb the motion control system, while at the same time being difficult to model and compensate for. The reason for this is the complexity of the interaction forces and the uncertainty in estimating their real contribution [5], [6].

Microrobotics research ranges from fundamental force/motion principles [7] to real (future) applications such as targeted drug delivery [1] and industrial automation solutions [8]. Unfortunately much research mainly provides a proof of concept for an actuated microrobotic device by simply moving an agent in an open environment with little disturbance over a predefined path. The real world however is

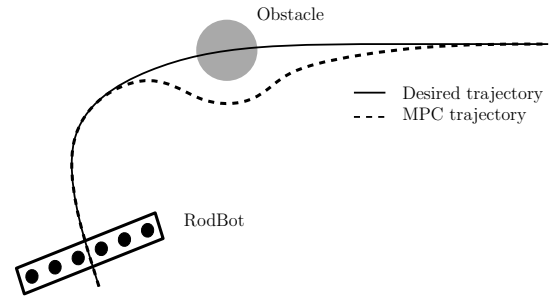


Fig. 1. The motion control problem is represented by trajectory tracking and obstacle avoidance while at the same time considering the constraints on the system and its inputs. A model predictive controller (MPC) is developed that solves this with a single cost function optimization routine.

more complex than this, as paths might be blocked, actuation and environmental constraints will limit the motion system and the actual task usually consists of more than just moving the actuated micro device.

We consider the motion control problem as represented in Fig. 1. A microrobot, the RodBot, must follow a desired trajectory (solid line) while avoiding obstacles in its environment (e.g. dirt or air bubbles, depicted as a grey circle). The RodBot’s rolling motion is nonholonomic and limited by kinematic constraints. This effectively means that it cannot move directly sideways and cannot rotate on the spot without a forward velocity. Moreover, the environmental conditions (low Reynolds number) and the actuation limits enforce a minimum and maximum forward and rotational velocity of the robot. The control system should take all these constraints into account and generate motion as represented by the dashed trajectory. With model predictive control (MPC) the optimization problem, which computes the optimal control action for a finite horizon, is subject to these constraints and limitations. In this work we present the development of a discrete-time, linearized model of the RodBot, define the constraints due to the system and the environment, and present the model predictive control framework that solves the motion control problem. Simulation and experimental results show that all constraints are taken into account and that obstacles can be avoided with MPC.

### A. Related work

Much research is carried out in the field of microrobotics. A general overview of planning and control in micromanipulation can be found in [9]. In this, a separation is made between microrobotic actuation, navigation and feedback control. MPC

This work is funded by the European Research Council Advanced Grant “Microrobotics and Nanomedicine (BOTMED)”.

Roel Pieters (corresponding author), Sievi Lombriser and Bradley J. Nelson are with the Institute of Robotics and Intelligent Systems, ETH Zurich, Tannenstrasse 3, Zurich, 8092, Switzerland, rpieters@ethz.ch, bnelson@ethz.ch

Alejandro Alvarez-Aguirre is with the Mechatronic System Design Group, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands, a.alvarez.aguirre@ieee.org

combines these into a single cost function optimization routine. Other work considering MPC for microrobotic motion control is as follows. In [10] generalized predictive control (GPC) was used to control the motion of a microbead driven by dielectrophoresis force. Predictive control is of particular interest due to the non-linearity and the high dynamics of the system, which induces high speed motion of the bead compared to the low image capture speed. Results are shown with simulations for the positioning of the microbead on a 2D plane. In [11] an external magnetic field is used to control the direction of motion of live microbial cells (*T. pyriformis*). A MPC approach is taken to make the cells follow a reference track. While experimental results show that tracking can be achieved, this also demonstrates the complexity of controlling live biological systems. More recent results are shown in [12] where multiple cells are to be controlled simultaneously while avoiding an obstacle. Experimental results however are shown with wheeled mobile robots and not the live microbial cells. A final similar approach on GPC is proposed in [13] and [14], where a ferromagnetic microcapsule is actuated by a MRI system for endovascular navigation. The controller however is proposed for navigation along a 2D path in the presence of pulsative flow as major disturbance and only simulation results are presented.

While all three approaches exemplify the benefits of predictive control, they only take the position error into account for optimization, with additional constraints on the system or its motion mostly being ignored. In this work, we present a MPC approach that deals with the constraints of the microrobotic system, the limitations of the actuation system and the disturbances due to the environment. The microrobot itself was introduced in [15] and analysed in [6]. This latter work showed that with a model-free controller, satisfactory tracking performance can be achieved (i.e. 50  $\mu\text{m}$  peak error for a 300  $\mu\text{m}$  wide robot). In this work we achieve a similar result. A performance comparison to other systems is difficult to make due to differences in actuation technique, motion generation and even size and shape of the microrobot. More importantly, the main contribution of this work is the integration of all mentioned constraints for trajectory tracking while avoiding obstacles.

In the remainder of this article, we present in Section II the MPC framework, the linearized and discretized model of the microrobot, as well as the constraints MPC has to take into account. Section III describes the microrobot, the magnetic control system and the practical implementation. Results are presented in Section IV in the form of simulations and experiments. In Section V we conclude the approach.

## II. MODEL PREDICTIVE CONTROL

MPC is an advanced control technique that uses on-line optimization performed over a receding horizon to calculate the best control actions. While the minimization of the cost function takes a finite receding horizon into account, only the first step of the control strategy is implemented. After that the optimization routine is re-evaluated [16]. The advantage that MPC has over traditional control techniques is the integration

of constraints and limitations of the system into a single controller. This enables trajectory tracking, obstacle avoidance and bounds on actuation to be combined in a single cost function optimization routine. Similar work regarding MPC for 1:43 scale RC race cars and for a group of unicycle mobile robots can be found in [17] and [18], respectively.

Following we introduce the linearized discrete-time model of the RodBot and the constrained cost function optimization problem for trajectory tracking and obstacle avoidance.

### A. Discrete-time Model of the RodBot

We define the state vector of a system as  $\mathbf{x}(t)$  and its input as  $\mathbf{u}(t)$ . With this we can then write the steady-state equations of the system as  $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$ . Similarly, the reference state and input are defined as  $\mathbf{x}_r(t)$  and  $\mathbf{u}_r(t)$ , respectively. The deviation from the reference values is defined as  $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_r(t)$  and  $\tilde{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}_r(t)$ .

As the RodBot is a nonlinear dynamical system it is first linearized about the reference, a concept also used for trajectory control with MPC in [17]. The linearized system of equations can be expressed as

$$\dot{\tilde{\mathbf{x}}}(t) = \mathbf{A}(t)\tilde{\mathbf{x}}(t) + \mathbf{B}(t)\tilde{\mathbf{u}}(t). \quad (1)$$

Due to the robot's design and the method of actuation (i.e. a rolling motion induces a forward velocity), the RodBot is limited to non-holonomic motion corresponding to motion of a unicycle mobile robot. Differences, however, include the inability to rotate on the spot and the drag force and drag torque due to the low-Reynolds number liquid environment. Moreover, a step-out frequency exists where the rotating RodBot cannot keep up with the input to the system (i.e., rotation frequency of the magnetic field). All these differences make it necessary to include constraints into MPC, in addition to the kinematic model of a unicycle

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = \begin{bmatrix} v_R \cos \theta_R \\ v_R \sin \theta_R \\ \omega_R \end{bmatrix}, \quad (2)$$

where the inputs of the system,  $v_R(t)$  and  $\omega_R(t)$ , are the forward and steering velocity (see Fig. 2). The variables  $x_R(t)$  and  $y_R(t)$  describe the position of the RodBot in the plane and  $\theta_R(t)$  its orientation.

Linearization and discretization of (2) leads to the exact discrete-time model of the RodBot as

$$\begin{bmatrix} \tilde{x}_{i+1} \\ \tilde{y}_{i+1} \\ \tilde{\theta}_{i+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & -T_s v_{r,i} \sin \theta_{r,i} \\ 0 & 1 & T_s v_{r,i} \cos \theta_{r,i} \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \\ \tilde{\theta}_i \end{bmatrix} + \underbrace{\begin{bmatrix} T_s \cos \theta_{r,i} & -\frac{1}{2} T_s^2 v_{r,i} \sin \theta_{r,i} \\ T_s \sin \theta_{r,i} & \frac{1}{2} T_s^2 v_{r,i} \cos \theta_{r,i} \\ 0 & T_s \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} \tilde{v}_i \\ \tilde{\omega}_i \end{bmatrix}, \quad (3)$$

with  $T_s$  the sampling time of the control system. Exact discretization takes into account that the control input remains constant on the interval between two sampling instants

$t_i = t \in [iT_s, (i+1)T_s]$  as occurs for discrete-time control systems [19]. In the matrices  $\mathbf{A}$  and  $\mathbf{B}$  the forward velocity  $v_{r,i}$  is derived from a reference trajectory as

$$v_{r,i} = \sqrt{\left(\frac{x_{r,i+1} - x_{r,i}}{T_s}\right)^2 + \left(\frac{y_{r,i+1} - y_{r,i}}{T_s}\right)^2}. \quad (4)$$

Likewise, the reference steering velocity is derived as

$$\omega_{r,i} = \frac{\theta_{r,i+1} - \theta_{r,i}}{T_s}. \quad (5)$$

Both are used to define the constraints for the inputs in the optimization routine described in Section II-C.

### B. Optimization Problem

The MPC problem can be formulated as [20]

$$\min_{\tilde{\mathbf{u}}} \frac{1}{2} \tilde{\mathbf{x}}_{N_p}^T \mathbf{P} \tilde{\mathbf{x}}_{N_p} + \frac{1}{2} \sum_{k=i}^{i+N_p-1} \tilde{\mathbf{x}}_k^T \mathbf{Q} \tilde{\mathbf{x}}_k + \tilde{\mathbf{u}}_k^T \mathbf{R} \tilde{\mathbf{u}}_k \quad (6)$$

subject to:

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1} &= \mathbf{A}_k \tilde{\mathbf{x}}_k + \mathbf{B}_k \tilde{\mathbf{u}}_k & k \in \{i, \dots, i+N_p-1\}, \\ \tilde{\mathbf{x}}_k &\in \mathcal{X} & k \in \{i, \dots, i+N_p-1\}, \\ \tilde{\mathbf{x}}_{N_p} &\in \mathcal{X}_f, \\ \tilde{\mathbf{u}}_{min} &\leq \tilde{\mathbf{u}}_k \leq \tilde{\mathbf{u}}_{max} & k \in \{i, \dots, i+N_p-1\}, \\ \tilde{\mathbf{x}}_0 &= \tilde{\mathbf{x}}(0). \end{aligned}$$

with  $\mathbf{P} = \mathbf{P}^T \succeq \mathbf{0}$ ,  $\mathbf{Q} = \mathbf{Q}^T \succeq \mathbf{0}$ ,  $\mathbf{R} = \mathbf{R}^T \succ \mathbf{0}$ ,  $\tilde{\mathbf{u}}_{min} \in \mathbb{R}^2$ ,  $\tilde{\mathbf{u}}_{max} \in \mathbb{R}^2$ ,  $\mathcal{X} \subseteq \mathbb{R}^3$  and  $\mathcal{X}_f \subseteq \mathcal{X}$ .  $\mathbf{P}$ ,  $\mathbf{Q}$  and  $\mathbf{R}$  are full column rank matrices penalizing the entries in the optimization function. In particular,  $\mathbf{R}$  penalizes the input deviations,  $\mathbf{Q}$  penalizes the state deviations and  $\mathbf{P}$  penalizes the terminal state deviation of the control horizon.  $\mathcal{X}$  denotes the set of states from which the optimization solver is allowed to select values, and should be convex. A separate set is defined for the deviation of the final state  $\tilde{\mathbf{x}}_{N_p}$ , and marked as  $\mathcal{X}_f$ . This is necessary to ensure closed loop stability [20]. The optimization problem returns  $\tilde{\mathbf{u}} = [\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_{i+1}, \dots, \tilde{\mathbf{u}}_{i+N_p-1}]^T$ , where  $N_p$  is the prediction horizon.

### C. Constraint Definition

The optimization problem can be solved considering several equality and inequality constraints. The constraint in (6) that describes the model of the system is described by the matrices  $\mathbf{A}$  and  $\mathbf{B}$  defined in (3), and includes the nonholonomic properties of the system. This allows predicting states over the horizon by computing the next state from the current state and input.

The second constraint in (6) gives restrictions on the states. This includes bounding the position of the RodBot as imposed by the borders of the camera image and the avoidance of obstacles (e.g. dirt or air bubbles). The obstacle-free region, as defined by excluding circular obstacles, is non-convex. Therefore, the constraint on obstacle avoidance is included by restricting the position of the RodBot to lie only in one half of a plane as described by the reference and potential obstacles. Let the center of the  $j^{th}$  circle to be avoided be

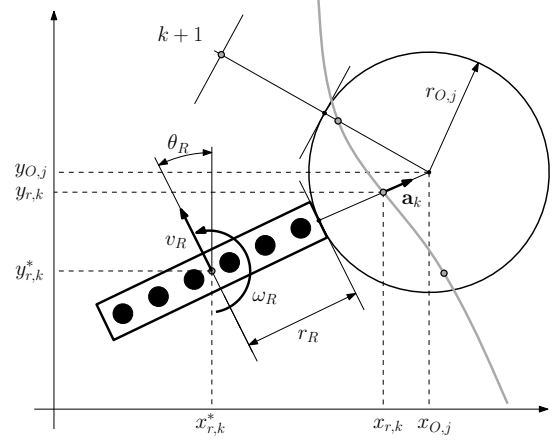


Fig. 2. Mechanism to avoid a circular obstacle: At each point of the reference path (gray), a half space is defined by connecting the reference point  $(x_{r,k}, y_{r,k})$  with the obstacle center  $(x_{O,j}, y_{O,j})$ . At the point where the resulting line touches the circle an additional distance of half the RodBot length  $r_R = l_R/2$  is added. This is the border of the half space at time  $k$ . The second set of lines (top) shows the same procedure at time  $k+1$ .

defined as  $(x_{O,j}, y_{O,j})$  and its radius as  $r_{O,j}$ . This radius is chosen such that interaction forces with nearby obstacles can be ignored. Instead of defining the circle as a forbidden region, the allowed region is defined as a half space that changes at each time step. A description of the half space is developed by defining the vector  $\mathbf{a}_k$  as a normal vector pointing from the reference point  $(x_{r,k}, y_{r,k})$  towards  $(x_{O,j}, y_{O,j})$ , the obstacle center (see Fig. 2). The half space is then given as the set [21]

$$\mathcal{Q}_k = \{\mathbf{q} \in \mathbb{R}^2 \mid \mathbf{a}_k^T \mathbf{q} \leq b_k\}, \quad (7)$$

where  $\mathbf{q} = [x_{r,k}^*, y_{r,k}^*]^T$  is a two-dimensional vector with the allowed future positions and where the normal vector  $\mathbf{a}_k$  is

$$\mathbf{a}_k = \frac{1}{\sqrt{(x_{O,j} - x_{r,k})^2 + (y_{O,j} - y_{r,k})^2}} \begin{bmatrix} x_{O,j} - x_{r,k} \\ y_{O,j} - y_{r,k} \end{bmatrix}, \quad (8)$$

and the scalar  $b_k$  is the distance between the reference point and the border of the half space

$$b_k = \sqrt{(x_{O,j} - x_{r,k})^2 + (y_{O,j} - y_{r,k})^2} - r_{O,j} - r_R, \quad (9)$$

where  $r_R = l_R/2$ , half the width of the RodBot. For every time step  $k \in \{i, \dots, i+N_p-1\}$ , the inequality (7) introduces another half space  $\mathcal{Q}_k$  such that  $\mathcal{X} \subseteq \mathcal{Q}_k \times \mathbb{R}$ .

The constraints in (6) on the inputs are imposed as lower and upper bounds on the forward and steering velocity of the RodBot and are chosen by the two-element vectors  $\tilde{\mathbf{u}}_{min} = [\tilde{v}_{min}, \tilde{\omega}_{min}]^T$  and  $\tilde{\mathbf{u}}_{max} = [\tilde{v}_{max}, \tilde{\omega}_{max}]^T$ . These constraints correspond to the limitations of the actuation system and to the low Reynolds number liquid environment. As the microrobot is submerged in a fluid the properties of this fluid (i.e., the viscosity) prescribe the amount of slip that occurs when rolling [15]. This affects the forward and steering velocity of the device. Moreover, as a rectangular cuboid the RodBot cannot rotate on the spot with zero forward velocity.

The last entry in (6) defines the initial state deviation as the current measured (or simulated) state deviation.

### III. EXPERIMENTAL SYSTEM

#### A. The RodBot

The microrobot, named the RodBot, is a wireless mobile device with a typical size of  $50 \times 60 \times 300 - 600 \mu m$ . The polymer body of the robot contains several soft-magnetic posts (perpendicular to the long axis) which align with an external magnetic field (see Fig. 3). When this field changes, the RodBot realigns with it. If the RodBot rolls without slipping, the forward velocity equals the rod perimeter times the rotational frequency. The deviation of the experimental data from this ideal forward velocity indicates slip of the RodBot. At higher forward velocity and higher viscosity of the surrounding liquid the fluidic drag force and drag torque increase, and the slip increases [15]. In low viscosity solutions, e.g. isopropanol, the maximum velocity of the RodBot is  $1 \text{ mm/s}$  ( $\approx 17$  body lengths/s). The RodBot rolls on a substrate in a low Reynolds number liquid environment (isopropanol) and generates flows to manipulate objects. In particular, objects can be manipulated by either a pushing strategy or a fluidic trapping technique. The RodBot was introduced in [15] and its capabilities was analysed in [6].

#### B. Magnetic Control System

The magnetic control system includes the eight-coil Magnetic Field Generator (MFG) as described in [22] and commercialized by MagnetbotiX AG<sup>1</sup>, a microscope with camera for visual feedback, and a container enclosing the RodBot (see Fig. 4). The system is controlled through C++ by a single computer running Ubuntu Linux and is capable of 5-DOF wireless control of micro- and nano-structures (3-DOF position, 2-DOF pointing orientation), within a spherical workspace with a diameter of approximately  $10 \text{ mm}$ . This allows for magnetic fields and field gradients up to  $20 \text{ mT}$  and  $2 \text{ T/m}$  at control frequencies up to  $4 \text{ kHz}$ . The motion of the RodBot is controlled at  $100 \text{ Hz}$  (magnetic field), where visual feedback returns the pose of the micro agent at  $30 \text{ Hz}$  as described in [6].

A rotating magnetic field enables a forward velocity  $v_R$  proportional to the rotational frequency of the magnetic field and the steering angle  $\theta_R$  corresponds to the orientation of the magnetic field (see Fig. 2). The magnetic field is generated by a set of static electromagnets at any desired point in the workspace as explained in [23].

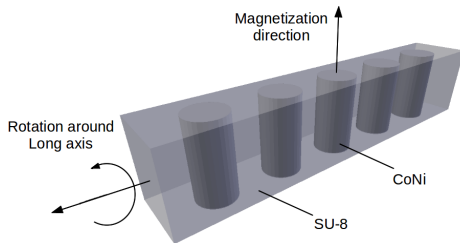


Fig. 3. The RodBot has a polymer body (SU-8) and soft-magnetic posts (CoNi) that align with an external magnetic field.

<sup>1</sup><http://www.magnetbotix.com>

#### C. MPC Solution with FiOrdOs

For the optimization routine the FiOrdOs code generator toolbox [21], [24] is used which accepts an optimization problem in the form of

$$\min_{\mathbf{z}} \left\{ \frac{1}{2} \mathbf{z}^T \mathbf{H} \mathbf{z} + c(\tilde{\mathbf{x}}_0) \mid \mathbf{A}_i \mathbf{z} \leq \mathbf{b}_i, \mathbf{A}_e \mathbf{z} = \mathbf{b}_e \right\}, \quad (10)$$

where  $\mathbf{z} = \left[ \tilde{\mathbf{u}}_0^T \tilde{\mathbf{u}}_1^T \dots \tilde{\mathbf{u}}_{N_p-1}^T \mid \tilde{\mathbf{x}}_1^T \tilde{\mathbf{x}}_2^T \dots \tilde{\mathbf{x}}_{N_p}^T \right]^T \in \mathbb{R}^{N_p(n_u+n_x)}$ ,  $\mathbf{H} = \text{diag}(\mathbf{R}, \mathbf{Q}, \mathbf{I})$ , and  $c(\tilde{\mathbf{x}}_0) = \frac{1}{2} \tilde{\mathbf{x}}_0^T \mathbf{Q} \tilde{\mathbf{x}}_0$ , where  $n_u$  and  $n_x$  is the size of the input and state vector, respectively. The matrices  $\mathbf{A}_e$  and  $\mathbf{b}_e$  describe the system and form an equality constraint for all future time steps. For obstacle avoidance the inequality constraints as described by (7) are passed to the solver in form of the matrix inequality  $\mathbf{A}_i \mathbf{z} \leq \mathbf{b}_i$ , which integrates (8) and (9) for all future time steps. As FiOrdOs cannot handle ellipsoids as sets for the inequality constraints a change of variables is made and the last state of the horizon in  $\mathbf{z}$  is replaced by  $\hat{\tilde{\mathbf{x}}}_{N_p} = \mathbf{P}^{1/2} \tilde{\mathbf{x}}_{N_p}$ . The change of variables is the reason for the identity matrix as the last block of  $\mathbf{H}$  [21]. In FiOrdOs the first-order primal-dual approach is used which can handle equality and/or inequality constraints, and where  $\mathbf{H} \succeq \mathbf{0}$ , and  $\mathbf{H} = \mathbf{0}$  is allowed [21]. This was used for both the simulation and experiments presented in the following section.

### IV. RESULTS

In order to evaluate the MPC framework several simulations and experiments are carried out. This includes the following of a loop-shaped trajectory and the following of a circular-shaped trajectory while avoiding a simulated circular obstacle. For the experiments a RodBot with 8 soft-magnetic posts and a width of  $l_R = 500 \mu m$  is used.

#### A. Loop Following

Fig. 5 shows the simulation and experimental results for the following of a loop-shaped trajectory with the RodBot. Fig. 6 shows the reference and measured forward and steering velocity for experiment 1. With  $\mathbf{P} = \mathbf{Q} = \text{diag}(1, 1, 1000)$  a higher penalty was given for the orientation deviation than

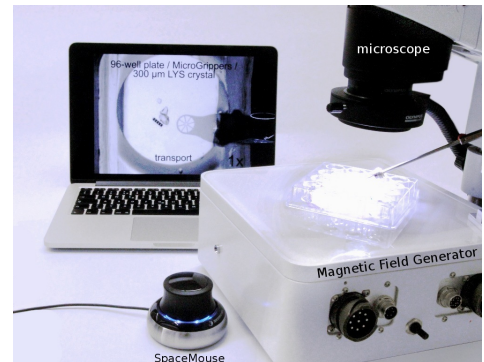


Fig. 4. The experimental system contains the eight-coil magnetic field generator (MFG, white box), a microscope with camera for visual feedback, and a laptop computer for processing.

for deviations in position. The constraints on the forward  $v_R$  and steering velocity  $\omega_R$  are considered as inequality constraints in  $\mathbf{z}$  with  $0 \text{ mm/s} \leq v_R \leq 1 \text{ mm/s}$ , and  $-\pi/8 \text{ rad/s} \leq \omega_R \leq \pi/8 \text{ rad/s}$  (red lines in Fig. 6), and therefore  $\mathbf{R} = \mathbf{0}_{2 \times 2}$ . The reference velocities stay within these bounds. Due to measurement noise the measured steering velocity rises slightly above the input constraint. These parameters are used for both the simulation and experimental results and, as shown, give a similar response. For the three experiments the average root mean square (RMS) error is  $87 \mu\text{m}$  with a standard deviation (SD) of  $57 \mu\text{m}$ . The peak lateral steady state error stays within  $160 \mu\text{m}$ .

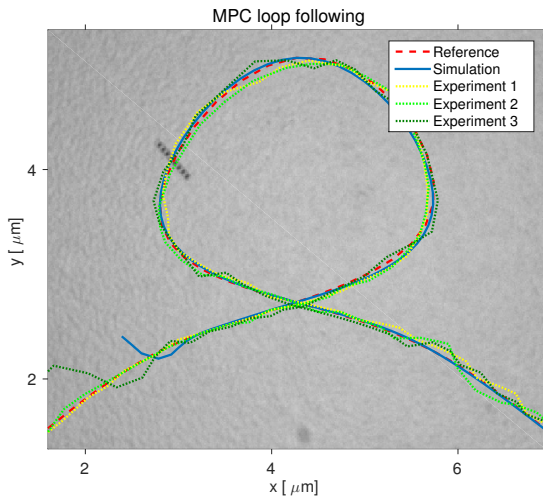


Fig. 5. Simulation (solid blue) and experimental (dotted) results for the following of a loop-shaped trajectory with MPC. The RodBot has 8 soft-magnetic posts and is  $500 \mu\text{m}$  wide. The penalty matrices are defined as  $\mathbf{P} = \mathbf{Q} = \text{diag}(1, 1, 1000)$ ,  $\mathbf{R} = \mathbf{0}_{2 \times 2}$ . For the three experiments the average root mean square (RMS) error is  $87 \mu\text{m}$  with a standard deviation (SD) of  $57 \mu\text{m}$ . The peak lateral steady state deviation stays within  $160 \mu\text{m}$ .

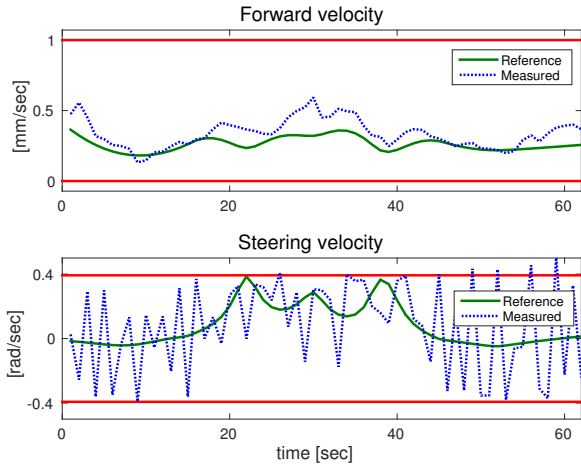


Fig. 6. Reference and measured velocities for experiment 1. The constraints on the input are defined as  $0 \text{ mm/s} \leq v_R \leq 1 \text{ mm/s}$ , and  $-\pi/8 \text{ rad/s} \leq \omega_R \leq \pi/8 \text{ rad/s}$  for the forward velocity and the steering velocity, respectively (solid red lines).

## B. Obstacle Avoidance

Fig. 7 shows the experimental results for the following of a circular trajectory with the RodBot while avoiding a static circular obstacle. The solid brown circles in the figure represent the boundary that the RodBot should avoid: the outer circle for the center of the RodBot, the inner circle for the outer limits of the RodBot. The executed motion of the center of the RodBot (experiment 5 and 6) stays outside of the obstacle's boundary. Experiment 4 (yellow) shows the RodBot slightly inside the obstacle region, explained by the fact that the real system gives a slightly different result than what the model can predict. To be sure to not collide with an obstacle, an additional security distance can be added to the circle radius. Moreover, oscillatory behaviour can be seen, which is likely due to the tight tuning between tracking performance and obstacle avoidance. This unwanted behaviour shows the complexity in tuning the MPC controller. Fig. 8 shows the reference and measured forward and steering velocity for experiment 6. For the circular trajectory the reference velocities  $v_R$  and  $\omega_R$  are constant. The same penalties as for the loop trajectory experiments are used. Due to measurement noise the measured steering velocity rises slightly above the input constraint. Outside the obstacle region the tracking performance is similar to the tracking performance of loop-following. During obstacle avoidance, the tracking error is 2 to 3 times larger. The reason for this is that the obstacle changes the reference position quite drastically, and, in order to comply with all constraints, the RodBot is forced to avoid the obstacle with more distance.

## C. Discussion

Because of the low velocity of the RodBot and the considerable damping due to the low Reynolds number fluid, the reaction to changing inputs is rather slow. Therefore, for all simulations and experiments the MPC controller was set to replan the trajectory at 1 Hz, and the prediction horizon was set to  $N_p = 4$ , enabling to 'look ahead' for four seconds. These results show that constraints on the forward and steering velocity are taken into account and that obstacles can be avoided with MPC.

## V. CONCLUSIONS

In this work we considered the motion control of a microrobot, the RodBot, submersed in a fluid and rolling on a supporting surface. As such the constraints the controller has to take into account are the nonholonomic properties of the microrobot, the limitations due to the actuation system and the low Reynolds number liquid environment, as well as the avoidance of potential obstacles on a desired trajectory. The task of trajectory tracking with aforementioned constraints is achieved by utilizing MPC. For this, a discrete-time, linearized model of the RodBot is developed and an optimization routine is defined to solve the motion control problem. Simulation and experimental results are presented with an 8-coil electromagnetic actuation system.

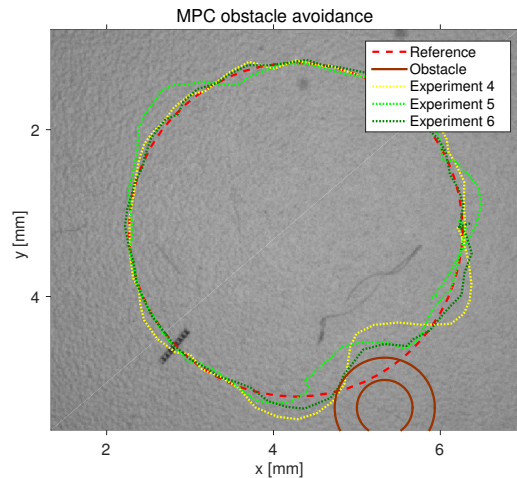


Fig. 7. Experimental results for the following of a circular trajectory while avoiding a static circular obstacle with MPC. The RodBot has 8 soft-magnetic posts and is  $500 \mu\text{m}$  wide. The penalty matrices are defined as  $\mathbf{P} = \mathbf{Q} = \text{diag}(1, 1, 1000)$ ,  $\mathbf{R} = \mathbf{0}_{2 \times 2}$ . The outer solid brown circle represents the boundary for the center of the RodBot, while the inner solid brown circle represents the boundary for the outer limits of the RodBot.

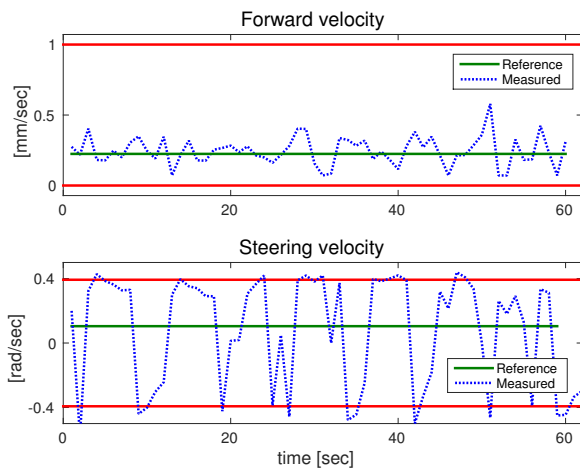


Fig. 8. Reference and measured velocities for experiment 6. The constraints on the input are defined as  $0 \text{ mm/s} \leq v_R \leq 1 \text{ mm/s}$ , and  $-\pi/8 \text{ rad/s} \leq \omega_R \leq \pi/8 \text{ rad/s}$  for the forward velocity and the steering velocity, respectively (solid red lines). By definition, the reference does not include obstacle avoidance.

## REFERENCES

- [1] B. J. Nelson, I. K. Kaliakatsos, and J. J. Abbott, "Microrobots for minimally invasive medicine," *Annual Review of Biomedical Engineering*, vol. 12, no. 1, pp. 55–85, 2010.
- [2] C.-J. Kim, A. Pisano, and R. Muller, "Silicon-processed overhanging microgripper," *Journal of Microelectromechanical Systems*, vol. 1, no. 1, pp. 31–36, 1992.
- [3] M. Kummer, J. Abbott, B. Kratochvil, R. Borer, A. Sengul, and B. Nelson, "Octomag: An electromagnetic system for 5-DOF wireless micromanipulation," *IEEE Transactions on Robotics*, vol. 26, no. 6, pp. 1006–1017, 2010.
- [4] W. Hu, K. S. Ishii, and A. T. Ohta, "Micro-assembly using optically controlled bubble microrobots," *Applied Physics Letters*, vol. 99, no. 9, p. 094103, 2011.
- [5] J. Abbott, Z. Nagy, F. Beyeler, and B. Nelson, "Robotics in the small, part I: Microbotics," *IEEE Robotics Automation Magazine*, vol. 14, no. 2, pp. 92–103, 2007.

- [6] R. S. Pieters, H.-W. Tung, S. Charreyron, S. F. Sargent, and B. J. Nelson, "Rodbot: a rolling microrobot for micromanipulation," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 4042–4047, 2015.
- [7] J. J. Abbott, M. Cosentino Lagomarsino, L. Zhang, L. Dong, and B. J. Nelson, "How should microrobots swim?," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1434–1447, 2009.
- [8] H.-W. Tung, D. F. Sargent, and B. J. Nelson, "Protein crystal harvesting using the rodbot - a wireless, mobile microrobot," *Journal of Applied Crystallography*, vol. 4, pp. 692–700, 2014.
- [9] A. Banerjee and S. Gupta, "Research in automated planning and control for micromanipulation," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 485–495, 2013.
- [10] M. Kharboutly, M. Gauthier, and N. Chaillet, "Predictive control of a micro bead's trajectory in a dielectrophoresis-based device," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 5616–5621, 2010.
- [11] Y. Ou, D. H. Kim, P. Kim, M. J. Kim, and A. Julius, "Motion control of tetrahymena pyriformis cells with artificial magnetotaxis: Model predictive control (MPC) approach," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 2492–2497, 2012.
- [12] Y. Ou, P. Kang, M. J. Kim, and A. Julius, "Algorithms for simultaneous motion control of multiple *T. pyriformis* cells: Model predictive control and particle swarm optimization," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 3507–3512, 2015.
- [13] K. Belharet, D. Folio, and A. Ferreira, "Endovascular navigation of a ferromagnetic microrobot using MRI-based predictive control," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2804–2809, 2010.
- [14] K. Belharet, D. Folio, and A. Ferreira, "Simulation and planning of a magnetically actuated microrobot navigating in the arteries," *IEEE Trans. on Biomedical Engineering*, vol. 60, no. 4, pp. 994–1001, 2013.
- [15] H.-W. Tung, K. E. Peyer, D. F. Sargent, and B. J. Nelson, "Noncontact manipulation using a transversely magnetized rolling robot," *Applied Physics Letters*, vol. 103, p. 114101, September 2013.
- [16] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer-Verlag, 2011.
- [17] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, 2014.
- [18] S. Adinandra, E. Schreurs, and H. Nijmeijer, "A practical model predictive control for a group of unicycle mobile robots," in *Proc. of the 4th IFAC Conf. on Nonlin. Model Pred. Control*, vol. 4, pp. 472–477, 2012.
- [19] C. Rabbath and N. Léchevin, *Discrete-Time Control System Design with Applications*. Springer-Verlag, 2013.
- [20] F. Borrelli, *Constrained Optimal Control of Linear & Hybrid Systems*, vol. 290. Springer-Verlag, 2003.
- [21] F. Ullmann, "FiOrdOs: A Matlab Toolbox for C-Code Generation for First Order Methods," Master's thesis, ETH Zurich, 2011.
- [22] S. Schürle, S. Erni, M. Flink, B. Kratochvil, and B. Nelson, "Three-dimensional magnetic manipulation of micro- and nanostructures for applications in life sciences," *IEEE Trans. on Magnetics*, vol. 49, no. 1, pp. 321–330, 2013.
- [23] B. E. Kratochvil, M. P. Kummer, S. Erni, R. Borer, D. R. Frutiger, S. Schürle, and B. J. Nelson, "Minimag: a hemispherical electromagnetic system for 5-DOF wireless micromanipulation," in *Experimental Robotics*, pp. 317–329, Springer, 2014.
- [24] D. Kufoalor, S. Richter, L. Imsland, T. Johansen, M. Morari, and G. Eikrem, "Embedded model predictive control on a PLC using a primal-dual first-order method for a subsea separation process," in *Proc. of 22nd IEEE Mediterranean Conference of Control and Automation (MED)*, pp. 368–373, 2014.