Control of Wall Mounting Robot

Sloth, Christoffer; Pedersen, Rasmus

Published in:
20th IFAC World Congress Toulouse, France, 9–14 July 2017, Proceedings

DOI (link to publication from Publisher):
10.1016/j.ifacol.2017.08.1113

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Sloth, C., & Pedersen, R. (2017). Control of Wall Mounting Robot. In 20th IFAC World Congress Toulouse, France, 9–14 July 2017, Proceedings (1 ed., Vol. 50, pp. 5648-5653). Elsevier. https://doi.org/10.1016/j.ifacol.2017.08.1113

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 5648–5653

Control of Wall Mounting Robot

Christoffer Sloth * Rasmus Pedersen *

* *Department of Electronic Systems, Automation and Control, Aalborg University, Denmark, (e-mail: {ces, rpe}@es.aau.dk).*

Abstract: This paper presents a method for designing controllers for trajectory tracking with actuator constraints. In particular, we consider a joystick-controlled wall mounting robot called WallMo. In contrast to previous works, a model-free approach is taken to the control problem, where the path parameter for the trajectory is adapted online. We demonstrate the method on a dynamic model of WallMo using predefined motion primitives, i.e. we consider path-constrained trajectory generation. It is seen that the control scheme results in improved trajectory tracking.

Keywords: Robotics technology, Motion Control Systems

1. INTRODUCTION

It is desirable to use robots in the construction of buildings, as robots can improve the work environment by alleviating heavy lifting, and possibly improve productivity. It is impractical to have caging around robots on construction sites to ensure safety. Therefore, this kind of robot is often controlled by a human operator; however, more sophisticated methods for designing safe collaborative robots also exist as explained in Zanchettin et al. (2016). We consider the robot WallMo that installs glass walls in office buildings, where a glass wall consists of several (2 m to 3.5 m tall) glass panels, with a mass of up to 130 kg. The installation should be accomplished with an accuracy of 1.5 mm. To obtain this accuracy a human operator gives joystick inputs to control the robot. To increase productivity, the robot should execute trajectories as fast as possible; thus, actuators may reach their limits during the execution of a motion. In this situation path-constrained time-optimal control should be applied.

Time-optimal motion planning is very important in the control of robots, as task execution time is a main parameter for improving the productivity and profitability of robots, according to Constantinescu and Croft (2000). Time-optimal motion planning has been addressed in numerous papers such as von Stryk and Bulirsch (1992); Gasparetto et al. (2015). The problem is often decomposed into two subproblems: path planning and trajectory planning, as described by Verscheure et al. (2009). However, this paper only addresses trajectory planning.

The papers Chen and Desrochers (1989); Pfeiffer and Johanni (1987) address path-constrained time-optimal control, where actuator constraints and other limits are taken into account. To solve this problem, the dynamical model of the system is exploited, by parameterizing it in the path parameters (a path parameter parameterizes a path). The same problem is addressed in Verscheure et al. (2009); however, in that paper a convex optimization problem is formulated to efficiently solve the problem.

To incorporate dynamic constraints in the trajectory planning, a dynamic scaling of trajectories may be employed (Siciliano et al., 2009, Section 7.7). This method can be used for scaling a given trajectory such that dynamic constraints are not violated. This method exploits the dynamic model of the robot, and relies on a predefined parametrization of the scaling function.

The work by Akrour et al. (2016) considers the trajectory optimization problem via reinforcement learning, and does therefore not need a model of the system. However, it relies on being able to execute the system multiple times to obtain an acceptable trajectory.

The considered robot is controlled along pre-specified paths, where a joystick gives a reference for the time derivative of the path parameter. Thus, a trajectory is not followed along a path with a specified end-point; in addition, the dynamic parameters of the robot and the load on the robot are unknown. Therefore, the trajectory should be generated online and adapted according to the current operating conditions. This is similar to standard constraint handling in anti-integrator windup as explained in Åström and Hägglund (2006).

This paper presents a methods for minimizing the execution time of path-constrained motions. We take a model-free approach to the problem (only the kinematic model is utilized), since the parameters describing the dynamics of the robot are uncertain, and the mass of the load is unknown. The robot is controlled via speed controllers on each joint with an operational space control, where the reference speed is given by a joystick. A switched control law is designed to maximize the speed in constrained operation. The work is similar to Constantinescu and Croft (2000); however, this paper does not rely on the model, and does not allow the control of the torque directly.

The paper is organized as follows. Section 2 provides a problem formulation and sketches how a glass panel can be installed with a robot; subsequently, a model of the considered robot called WallMo is given in Section 3. Section 4 presents motion primitives that define the path, which the tool of the robot must follow. The main contribution of the paper is given in Section 5; this is a method for handling actuator constraint and maximizing the speed of a motion.

10.1016/j.ifacol.2017.08.1113

Finally, simulation results are presented in Section 6, and conclusions are provided in Section 7.

2. PROBLEM FORMULATION

The purpose of this section is to present the problem addressed in this paper. The aim is to design a control system for a robot that should install glass panels (up to 3.5 m tall). The glass panels have to be mounted in 15 mm wide rails (the glass panels are 12 mm thick); thus, the robot should follow the designated path very accurately. The scenario and the robot are shown in Fig. 1.

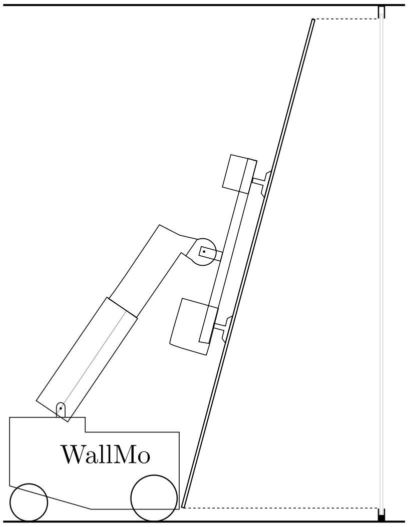


Fig. 1. Illustration of the wall mounting robot (WallMo) carrying a glass panel. The glass must be installed in the rails in the top and bottom.

The robot is controlled via a joystick throughout the installation of the glass. This task is difficult to accomplish, by controlling the joints individually; thus, some motion primitives should be defined, to obtain an intuitive user interface.

Problem 1. Define motion primitives that eases the operation of the robot from a joystick.

A solution to this problem is presented in Section 4.

The main problem addressed in this paper is to design a control system for following a path accurately, despite actuator constraints. In addition, the control system should work without knowledge about the dynamics of the robot.

Problem 2. Design a control system that maximizes the motion speed for path-constrained control of a robot with unknown dynamics and actuator constraints.

A solution to Problem 2 is presented in Section 5.

3. MODELING

This section presents the kinematic and dynamic models of the 3-joint arm of WallMo, which is used for controller design and validation. The base of the robot is stationary during installation; consequently, it is not considered in the remainder of the paper.

3.1 Kinematic Model

The kinematic model is described using the modified Denavit Hartenberg (DH) representation, see Craig (2005). The DH parameters for WallMo are provided in Table 1 and a visualization of the local coordinate systems is provided in Fig. 2. The manipulator has three degrees of freedom, and its generalized coordinates are defined as $q = [q_1\ q_2\ q_3]^{\mathrm{T}}$, where q_1 and q_3 are revolute joints and q_2 is a prismatic joint. The tool frame is defined to be at the top of the glass panel, aligned with the glass.

The considered robot is planar; thus, all the following transformation matrices are planar. The constant transformation from the last actuated joint q_3 to the tool frame is given by

$$ {}^{3}_{e}T = \begin{bmatrix} 1 & 0 & \delta_x \\ 0 & 1 & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

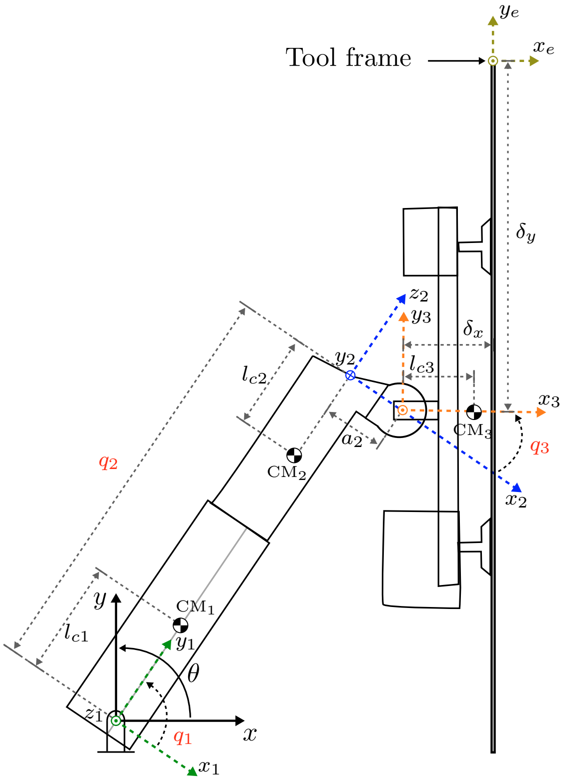


Fig. 2. Arm of WallMo with local coordinate systems and center of mass (CM) points defined. The generalized coordinates of the robot (q_1, q_2, q_3) are marked by red text color.

i	a_{i-1} [m]	d_i	θ_i	α_{i-1} [rad]	offset [rad]
1	0	0	q_1	0	$-\pi/2$
2	0	q_2	0	$-\pi/2$	0
3	a_2	0	q_3	$\pi/2$	0

Table 1. Link parameters of WallMo given in the modified DH representation, where q_i represents the i^{th} generalized coordinate.

The kinematic model, giving a relation between the tool frame and the base frame, is given by the following transformation matrix

$$
\begin{aligned}
{}^0_eT(q) &= {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_eT && (2)\\
&= \begin{bmatrix} s_{13} & c_{13} & a_2s_1 + \delta_y c_{13} + \delta_x s_{13} + c_1q_2 \\ -c_{13} & s_{13} & -a_2c_1 - \delta_x c_{13} + \delta_y s_{13} + s_1 q_2 \\ 0 & 0 & 1 \end{bmatrix}, && (3)
\end{aligned}
$$

where c_i (s_i) denotes $\cos(q_i)$ ($\sin(q_i)$) and c_{ij} (s_{ij}) denotes $\cos(q_i + q_j)$ ($\sin(q_i + q_j)$). This means that the forward kinematics are given by the following map

$$
K : (q_1, q_2, q_3) \mapsto \begin{bmatrix} a_2s_1 + \delta_y c_{13} + \delta_x s_{13} + c_1 q_2 \\ -a_2 c_1 - \delta_x c_{13} + \delta_y s_{13} + s_1 q_2 \\ q_1 + q_3 - \dfrac{\pi}{2} \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}
$$

The control presented in Section 5 exploits the inverse kinematics for defining path parameters; thus, it is given in the following

$$
\begin{aligned}
q_2 &= \sqrt{\tilde{x}^2 + \tilde{y}^2 - a_2^2} \qquad [\mathrm{m}] && (4a)\\
q_1 &= \operatorname{asin}\left(\frac{q_2\tilde{y} + a_2\tilde{x}}{q_2^2 + a_2^2}\right) \qquad [\mathrm{rad}] && (4b)\\
q_3 &= \frac{\pi}{2} + \theta - q_1 \qquad [\mathrm{rad}] && (4c)
\end{aligned}
$$

where

$$
\begin{aligned}
\tilde{x} &= x - \delta_y \cos\left(\frac{\pi}{2} + \theta\right) - \delta_x \sin\left(\frac{\pi}{2} + \theta\right)\\
\tilde{y} &= y - \delta_y \sin\left(\frac{\pi}{2} + \theta\right) + \delta_x \cos\left(\frac{\pi}{2} + \theta\right).
\end{aligned}
$$

Note that $q_2 > 0$ from Table 3 and that the range of q_1 results in a unique solution to this angle.

3.2 Dynamic Model

To simulate the robot, and study the proposed control law in a dynamic setting, a dynamic model is presented in this section.

We derive a model on the following standard form for an n degrees of freedom manipulator

$$
\tau = M(q)\ddot{q} + V(q, \dot{q}) + g(q) + F(\dot{q}) \qquad [\mathrm{Nm}] \qquad (5)
$$

where τ is a vector of joint torques, $q \in \mathbb{R}^n$ is a vector of generalized coordinates, $M(q)$ is the mass matrix, $V(q, \dot{q})$ is a vector of centrifugal and Coriolis terms, $g(q)$ is a vector of gravity terms, and $F(\dot{q})$ is a vector of friction terms. The parameter values of the model are found in the Appendix.

Several constraints are imposed on the system. These are bounds on the joint torques and joint angles. The bounds are defined as

$$
\underline{\tau}_i \leq \tau_i \leq \overline{\tau}_i \quad \text{and} \quad \underline{q}_i \leq q_i \leq \overline{q}_i
$$

for $i = 1, \ldots, n$ where $\underline{\tau}_i, \underline{q}_i$ and $\overline{\tau}_i, \overline{q}_i$ are lower and upper bounds. Values for the bounds are provided in the Appendix.

4. MOTION PRIMITIVES

This section introduces the three motion primitives defined to ease the installation of glass panels, giving a solution to Problem 1. We describe the three motion primitives, show how they ease the installation procedure, and parameterize their associated paths which are used for the control of the robot.

The proposed motion primitives are called *Move*, *Slide*, and *Lift*. The idea behind introducing motion primitives is to ease the manual installation of glass panels. The robot is controlled via the joystick shown in Fig. 3; thus, it is desired to minimize the number of modes and the number of mode switches during the installation of a glass panel. The motion primitives are illustrated in Fig. 4.

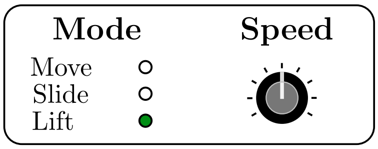


Fig. 3. Sketch of the utilized joystick, with three modes *Move*, *Slide*, and *Lift*, and a knob for adjusting the speed of the selected motion.

The motion *Move* will move the tool frame in the x-direction, and *Lift* moves the tool frame in the y-direction, while *Slide* rotates the glass panel, while having constant distance between the bottom of the glass panel and the ground, and constant x-coordinate of the top of the glass panel.

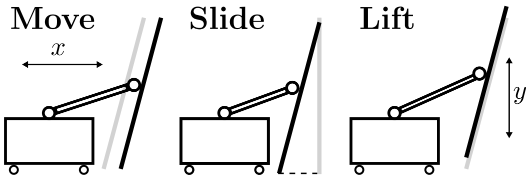


Fig. 4. Principle of the three motion primitives *Move*, *Slide*, and *Lift*.

By exploiting the motion primitives, a glass panel can be installed by going through a sequence of motions as shown in Fig. 5. In particular, starting with *Move* to place the top of the glass underneath the top rail, by the use of *Slide* to put the glass into the top rail. Finally, when the glass is vertical, it can be lowered into the bottom rail using *Lift*.

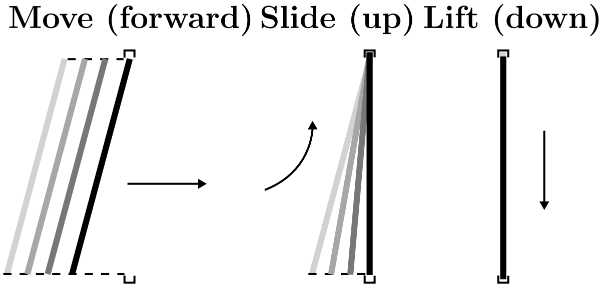


Fig. 5. Three step installation procedure, where each movement starts at the brightest color and ends at the darkest color.

We denote the configuration of the tool frame by $x_{\mathrm{e}} = (x, y, \theta)$ (in the base frame). Paths for the three motion primitives from the initial configuration of the tool frame $x_{\mathrm{e},0} = (x_0, y_0, \theta_0)$ are given in the following, where each motion primitive is parameterized in the path parameter $s \in \mathbb{R}$

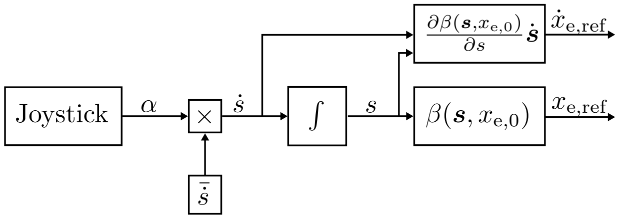
$$
\begin{aligned}
\beta_{\mathrm{Move}} &: (s, x_{\mathrm{e},0}) \mapsto (x_0 + s, y_0, \theta_0)\\
\beta_{\mathrm{Slide}} &: (s, x_{\mathrm{e},0}) \mapsto (x_0, y_0 + 2\delta_y(\cos(\theta_0 + s) - \cos(\theta_0)), \theta_0 + s)\\
\beta_{\mathrm{Lift}} &: (s, x_{\mathrm{e},0}) \mapsto (x_0, y_0 + s, \theta_0).
\end{aligned}
$$

5. CONTROLLER DESIGN

The purpose of this section is to present a trajectory generation that enables the motion primitives presented in Section 4 to be executed at maximal speed. This is accomplished by taking into account actuator limitations. Initially, the nominal control structure is presented, and the addressed challenge of actuator saturation is presented. Subsequently, a trajectory generation is presented that handles actuator constraints.

5.1 Nominal Control Structure

The considered robot is controlled via a joystick that generates the reference speed for a motion, $\dot{s}$. Thus, the output of the joystick is $\alpha \in [-1, 1]$ (in addition to a selection of the desired motion primitive) from which the reference speed of the movement is calculated as $\dot{s} = \alpha\bar{\dot{s}}$, where $\bar{\dot{s}} \in \mathbb{R}_+$ is the maximum reference speed of the movement. A schematic of the reference generation is shown in Fig. 6, where the index on the map β defining the desired path is omitted to shorten the notation.



Joystick
α
$\times$
$\dot{s}$
$\bar{\dot{s}}$
$\int$
s
$\frac{\partial\beta(\boldsymbol{s}, x_{\text{e},0})}{\partial s}\dot{\boldsymbol{s}}$
$\dot{x}_{\text{e,ref}}$
$\beta(\boldsymbol{s}, x_{\text{e},0})$
$x_{\text{e,ref}}$

Fig. 6. Schematic of reference generation, where the map β defines the motion primitive, and $\bar{\dot{s}}$ is the maximum speed of the motion.

Formally, the reference signals are computed as follows

$$x_{\text{e,ref}} = \beta(s, x_{\text{e},0})$$
$$\dot{x}_{\text{e,ref}} = \frac{\partial\beta(s, x_{\text{e},0})}{\partial s}\dot{s},$$

where the map $\beta(\cdot, x_{\text{e},0})$ is differentiable.

The computed reference $(x_{\text{e,ref}}, \dot{x}_{\text{e,ref}})$ is input to an operational space control of the robot, inspired by (Siciliano and Khatib, 2008, Section 8.6), with both feedforward (using $\dot{x}_{\text{e,ref}}$) and feedback (using $x_{\text{e,ref}}$). A schematic of the control system is shown in Fig. 7.

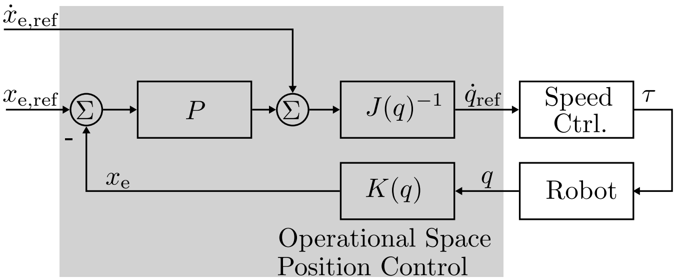


Fig. 7. Proposed position control system for WallMo, where $J(q)$ is the Jacobian matrix and P is a matrix with controller gains.

The operational space control shown in Fig. 7 tracks the reference when actuators are not saturated, i.e., $\underline{\tau} \leq \tau \leq \overline{\tau}$, where $\underline{\tau}$ and $\overline{\tau}$ are bounds on the actuator torque. However, when an actuator is saturated, the motion cannot be followed. In the mode *Lift*, the tool frame is moved vertically. This sets different demands to the actuators depending on the configuration of the robot. This is exemplified in Fig. 8 that illustrates a desired trajectory in joint space, and indicates when the actuator for joint 2 is in saturation.

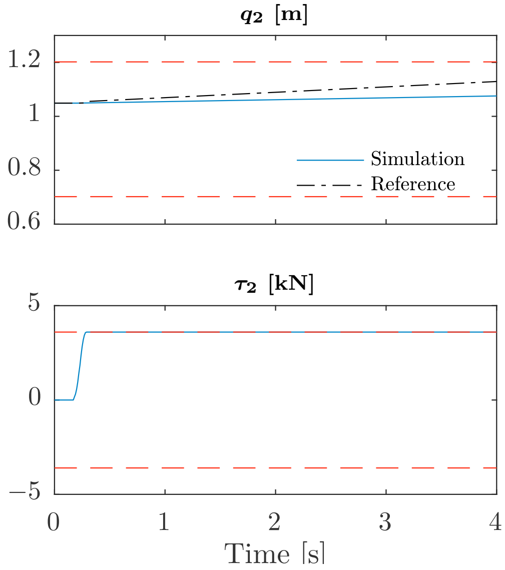


Fig. 8. Illustration of joint angle and joint torque for joint 2, when following a reference trajectory. The red dashed lines indicate the limits on q_2 and τ_2.

It is seen from Fig. 8 that the actuator for joint 2 quickly reaches saturation. As a consequence, the desired path is not followed as shown in Fig. 9. This undesired behavior must be corrected to give the system operator an intuitive interface for controlling the system.

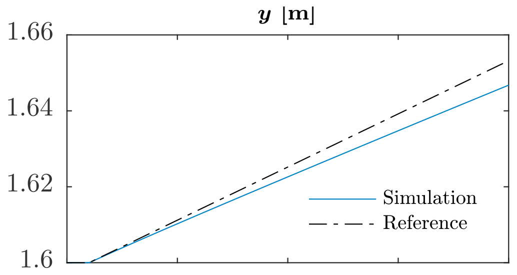


Fig. 9. Illustration of reference trajectory and the configuration of the tool frames y coordinate.

The next subsection presents a solution to this issue, by modifying the reference speed for the path parameter of the motion primitive online.

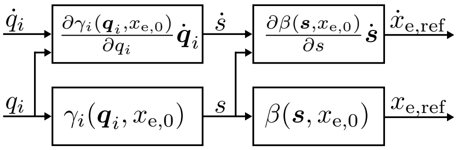
5.2 Trajectory Generation with Actuator Constraints

This section provides a method for online design of trajectories for systems with actuator saturation. Consequently, a solution is provided to the problem identified in the previous section.

We propose to parameterize the desired motion primitive in the generalized coordinate of the saturated joint, and let this joint define the speed of the motion when the system is in saturation. In particular, we define functions from the generalized coordinates q_i to the path parameter s for $i = 1, \dots, n$

$$\gamma_i : (q_i, x_{\mathrm{e},0}) \mapsto s.$$

The functions $\gamma_i(\cdot, x_{\mathrm{e},0})$ should be bijective and differentiable. As an example, when the torque constraint on joint i reaches saturation, i.e. $\tau_i = \overline{\tau}_i$, the trajectory generation is defined as shown in Fig. 10, and the input to joint i is $\tau_i = \overline{\tau}_i$. A description of trajectory generation is presented in the following.



$\dot{q}_i$
$\frac{\partial\gamma_i(\boldsymbol{q}_i, x_{\mathrm{e},0})}{\partial q_i}\dot{\boldsymbol{q}}_i$
$\dot{s}$
$\frac{\partial\beta(\boldsymbol{s}, x_{\mathrm{e},0})}{\partial s}\dot{\boldsymbol{s}}$
$\dot{x}_{\mathrm{e,ref}}$
q_i
$\gamma_i(\boldsymbol{q}_i, x_{\mathrm{e},0})$
s
$\beta(\boldsymbol{s}, x_{\mathrm{e},0})$
$x_{\mathrm{e,ref}}$

Fig. 10. Schematic of reference generation, when joint i has reached saturation.

It is assumed that the joystick input signal $\alpha : \mathbb{R} \to [-1, 1]$ is continuous. To provide a description of the trajectory generation, the set of actuators in saturation is defined as follows

$$I_{\mathrm{sat}}(t) = \{i \in \{1, \dots, n\} |\ \tau_i(t) \leq \underline{\tau}_i \text{ or } \tau_i(t) \geq \overline{\tau}_i\} \cup \{0\}.$$

The element $\{0\}$ is added to $I_{\mathrm{sat}}(t)$ to ease the subsequent notation. Next, we identify the index of the saturated actuator or joystick input that gives the slowest trajectory

$$i^*(t) = \arg \min_{i \in I_{\mathrm{sat}}(t)} \dot{s}_i. \tag{6}$$

where

$$\dot{s}_i = \begin{cases} \alpha(t)\overline{\dot{s}} & \text{for } i = 0 \\ \frac{\partial \gamma_i}{\partial q_i}(q_i(t), x_{\mathrm{e},0})\dot{q}_i(t) & \text{otherwise} \end{cases} \tag{7}$$

Finally, the path parameter is given by

$$s(t) = \int_{t^*}^{t} \dot{s}_0 dt + \gamma_{i^*}(q_{i^*}(t^*), x_{\mathrm{e},0}) \tag{8}$$

where

$$t^* = \max\{t \geq t_0 \mid\ I_{\mathrm{sat}}(t) \setminus \{0\} \neq \emptyset\} \tag{9}$$

and the path velocity is

$$\dot{s}(t) = \dot{s}_{i^*(t)}. \tag{10}$$

From the above equations it is seen that the trajectory generation is identical to the nominal one, when no actuators are saturated. However, when an actuator is saturated, then by (6) and (7) the speed of the path parameter is reduced to comply with the constraints of the saturated actuator, and by (8) the path parameter is given by the saturated joint.

6. SIMULATION OF LIFT MOTION

The purpose of this section is to exemplify the trajectory generation presented in Section 5 on the *Lift* motion on WallMo. We show that the tracking ability of the controller is improved compared to the nominal controller shown in Fig. 9. Control parameters are provided in the Appendix. To realize the proposed trajectory generation, the functions β_{Lift} and $\gamma_{\mathrm{Lift},i}$ and their partial derivatives need to be found. The motion primitive for *Lift* is defined in Section 4 as $\beta_{\mathrm{Lift}} : (s, x_{\mathrm{e},0}) \mapsto (x_0, y_0 + s, \theta_0)$; thus, we have

$$\frac{\partial \beta_{\mathrm{Lift}}}{\partial s} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

The inverse kinematics is needed to find the functions $\gamma_{\mathrm{Lift},i}$ for $i = 1, 2, 3$. It is seen from (4) that the path parameter associated to the *Lift* motion can be parameterized in the generalized coordinates q_1, q_2, q_3 as follows

$$\gamma_{\mathrm{Lift},1} : (q_1, x_{\mathrm{e},0}) \mapsto \frac{s_1\tilde{x}_0 - a_2}{c_1} - \tilde{y}_0$$
$$\gamma_{\mathrm{Lift},2} : (q_2, x_{\mathrm{e},0}) \mapsto \pm\sqrt{q_2^2 + a_2^2 - \tilde{x}_0^2} - \tilde{y}_0$$
$$\gamma_{\mathrm{Lift},3} : (q_3, x_{\mathrm{e},0}) \mapsto \frac{\sin(\pi/2 + \theta_0 - q_3)\tilde{x}_0 - a_2}{\cos(\pi/2 + \theta_0 - q_3)} - \tilde{y}_0.$$

where

$$\tilde{x}_0 = x_0 - \delta_y \cos\left(\frac{\pi}{2} + \theta_0\right) - \delta_x \sin\left(\frac{\pi}{2} + \theta_0\right)$$
$$\tilde{y}_0 = y_0 - \delta_y \sin\left(\frac{\pi}{2} + \theta_0\right) + \delta_x \cos\left(\frac{\pi}{2} + \theta_0\right)$$

From Table 3, it is seen that $c_1 \neq 0$ for the considered robot; thus, the singularities of $\gamma_{\mathrm{Lift},1}$ and $\gamma_{\mathrm{Lift},3}$ are not reached. Also, we restrict the robot such that $\tilde{y} > 0$, which implies that the expression for $\gamma_{\mathrm{Lift},2}$ is also unique.

Finally, the partial derivatives are given as

$$\frac{\partial \gamma_{\mathrm{Lift},1}}{\partial q_1} = \frac{s_1(\tilde{x}_0 s_1 - a_2)}{c_1^2} + \tilde{x}_0$$
$$\frac{\partial \gamma_{\mathrm{Lift},2}}{\partial q_2} = \frac{q_2}{\sqrt{q_2^2 + a_2^2 - \tilde{x}_0^2}}$$
$$\frac{\partial \gamma_{\mathrm{Lift},3}}{\partial q_3} = \frac{\sin(\frac{\pi}{2} + \theta_0 - q_3)(\tilde{x}_0 \sin(\frac{\pi}{2} + \theta_0 - q_3) - a_2)}{\cos(\pi/2 + \theta_0 - q_3)^2} - \tilde{x}_0$$

The above functions completely specifies the trajectory generation for the proposed control. A simulation of the *Lift* motion is shown in Fig. 11 and illustrates how the robot is now capable of following the referenced trajectory. It is seen from the figure that the tracking of the reference is good, as the speed of the motion has been scaled.

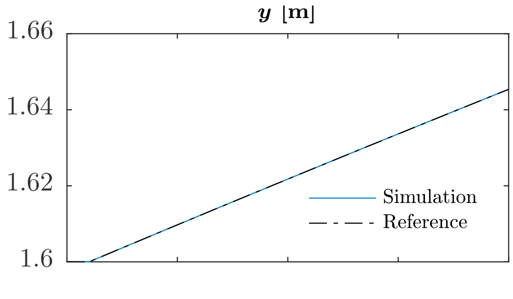


Fig. 11. Illustration of reference trajectory and the configuration of the tool frames y coordinate, when applying the proposed parameterization.

7. CONCLUSION

This paper introduced the wall mounting robot WallMo and presented a method for online trajectory design. A kinematic and dynamic model of the robot was provided, along with definitions of motion primitives for easing the installation procedure. The presented controller design is model-free; thus, it is appropriate for the control of systems with unknown dynamics and load. The control gives a point-wise maximal path velocity, despite saturation of actuators. The presented control structure ensured that the generated operational space trajectory was followed during run-time when no actuators were constrained. If actuators were constrained, it was shown how the trajectory could be changed online to improve the tracking of the motion. The effectiveness of the proposed method was illustrated through a simulation study.

APPENDIX

This appendix provides the parameters of the robot model and controllers.

Symbol	Explanation	Value	Unit
a_2		0.1520	m
δ_x	Displacement of panel	0.2262	m
δ_y	Elevation of panel	1	m

Table 2. Parameters for kinematics.

i	$\underline{q}_i$	$\bar{q}_i$	$\underline{\tau}_i$	$\bar{\tau}_i$
1	-0.1016 rad	1.3216 rad	-1 kNm	1 kNm
2	0.702 m	1.202 m	-3.6 kN	3.6 kN
3	0 rad	$\pi/2$ rad	-412 Nm	412 Nm

Table 3. Constraints on joint angles and joint torques.

Symbol	Explanation	Value	Unit
m_1, m_2	Mass link 1, 2	9	kg
m_3	Mass link 3 (+ glass)	90	kg
r_1	Radius link 1	0.217	m
r_2	Radius link 2	0.192	m
h_1	Height link 1	0.627	m
h_2	Height link 2	0.500	m
w_3	Width link 3	0.50	m
δ_3	Depth link 3	0.15	m
h_3	Height link 3	1	m
l_{c1}	Length to CM1	$h_1/2$	m
l_{c2}	Length to CM2	$h_2/2$	m
l_{c3}	Length to CM3	$2a_3/3$	m
$I_{1,11}$	x-inertia of link 1	$m_1(3r_1^2 + h_1^2)/12$	kg/m^2
$I_{1,22}$	y-inertia of link 1	$m_1(3r_1^2 + h_1^2)/12$	kg/m^2
$I_{1,33}$	z-inertia of link 1	$m_1 r_1^2/12$	kg/m^2
$I_{2,11}$	x-inertia of link 2	$m_2(3r_2^2 + h_2^2)/12$	kg/m^2
$I_{2,22}$	y-inertia of link 2	$m_2(3r_2^2 + h_2^2)/12$	kg/m^2
$I_{2,33}$	z-inertia of link 2	$m_2 r_2^2/12$	kg/m^2
$I_{3,11}$	x-inertia of link 3	$m_3(h_3^2 + \delta_3^2)/12$	kg/m^2
$I_{3,22}$	y-inertia of link 3	$m_3(w_3^2 + \delta_3^2)/12$	kg/m^2
$I_{3,33}$	z-inertia of link 3	$m_3(w_3^2 + h_3^2)/12$	kg/m^2
k_{f1}, k_{f2}	Friction joint 1, 2	2	-
k_{f3}	Friction joint 3	0.6	-

Table 4. Parameters for dynamic model. The model can be constructed using the procedure in Craig (2005).

Controller	Parameter	Value
Velocity PID Joint 1, 2	P, I, D	50e3, 50e6, 500
Velocity PID Joint 3	P, I, D	50e3, 50e6, 50
Operational Space	P	diag$(0.5, 0.5, 1)$

Table 5. Control parameters.

ACKNOWLEDGMENTS

This work is supported by Innovation Fund Denmark in project number 5150-00007A called WallMoBot.

REFERENCES

Akrour, R., Abdolmaleki, A., Abdulsamad, H., and Neumann, G. (2016). Model-free trajectory optimization for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*.

Åström, K.J. and Hägglund, T. (2006). *Advanced PID Control*.

Chen, Y. and Desrochers, A.A. (1989). Structure of minimum-time control law for robotic manipulators with constrained paths. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, 971–976 vol.2. doi:10.1109/ROBOT.1989.100107.

Constantinescu, D. and Croft, E.A. (2000). Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. *Journal of Robotic Systems*, 17(5), 233–249. doi:10.1002/(SICI)1097-4563(200005)17:5⟨233::AID-ROB1⟩3.0.CO;2-Y.

Craig, J.J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson Education International, 3rd edition.

Gasparetto, A., Boscariol, P., Lanzutti, A., and Vidoni, R. (2015). *Path Planning and Trajectory Planning Algorithms: A General Overview*, 3–27. Springer International Publishing. doi:10.1007/978-3-319-14705-5_1.

Pfeiffer, F. and Johanni, R. (1987). A concept for manipulator trajectory planning. *IEEE Journal on Robotics and Automation*, 3(2), 115–123. doi:10.1109/JRA.1987.1087090.

Siciliano, B. and Khatib, O. (eds.) (2008). *Springer Handbook of Robotics*. Springer Berlin Heidelberg. doi:10.1007/978-3-540-30301-5.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robotics: Modelling, Planning and Control*. Springer-Verlag London. doi:10.1007/978-1-84628-642-1.

Verscheure, D., Demeulenaere, B., Swevers, J., Schutter, J.D., and Diehl, M. (2009). Time-optimal path tracking for robots: A convex optimization approach. *IEEE Transactions on Automatic Control*, 54(10), 2318–2327. doi:10.1109/TAC.2009.2028959.

von Stryk, O. and Bulirsch, R. (1992). Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37(1), 357–373. doi:10.1007/BF02071065.

Zanchettin, A.M., Ceriani, N.M., Rocco, P., Ding, H., and Matthias, B. (2016). Safety in human-robot collaborative manufacturing environments: Metrics and control. *IEEE Transactions on Automation Science and Engineering*, 13(2), 882–893. doi:10.1109/TASE.2015.2412256.