

Set-based Control for Autonomous Spray Painting

Signe Moe¹, Jan T. Gravdahl² and Kristin Y. Pettersen¹

Abstract—In this paper, a method is presented for lowering the energy consumption and/or increasing the speed of a standard manipulator spray painting a surface. The approach is based on the observation that a small angle between the spray direction and the surface normal does not affect the quality of the paint job. Recent results in set-based kinematic control are utilized to develop a switched control system, where this angle is defined as a set-based task with a maximum allowed limit. Four different set-based methods are implemented and tested on a UR5 manipulator from Universal Robots. Experimental results verify the correctness of the method, and demonstrate that the set-based approaches can substantially lower the paint time and energy consumption compared to the current standard solution.

Note to Practitioners— This paper is motivated by the observation that uniform paint coating can be achieved even when the spray direction is not normal to the spray surface. By allowing a small orientation error, the manipulator maintains a constant spray velocity along the surface while following a more energy efficient trajectory at the end effector. By doing so, this velocity can also be increased as the proposed method requires less torque during the turns than the current industry standard. The method proposed in this paper is based on a kinematic control method and generates reference trajectories for the manipulator configuration and velocity every time-step which are inputs to the manipulator dynamic controller. If tracked, the references will allow the paint gun to follow the trajectory, including both straight line segments and turns, with constant velocity and thus achieve uniform paint coating. We show how to calculate the reference state of the manipulator and present the explicit expressions for solving and implementing the algorithms. The approach can also be used for other applications where an offset in the end-effector orientation improves performance, such as welding and 3D-printing.

I. INTRODUCTION

Robotic systems are often required to perform one or several tasks which are given in the operational space, for instance obtaining a certain desired end effector position and/or orientation. However, these systems are often controlled in joint space, and thus a variety of inverse kinematics algorithms have been developed to map desired behavior from the operational space to the joint space and thus generate reference trajectories for the joint controllers. The most common approach is to use a Jacobian-based method [1]-[3], such as the Jacobian transpose, damped least squares or pseudo-inverse. In particular, the pseudo-inverse Jacobian is also defined for systems that are not square nor have full

rank and is a widely used solution to the inverse kinematics problem [4]-[6].

A robotic system is said to be kinematically redundant if it possesses more degrees of freedom (DOFs) than those required to perform a certain task [7]. In this case, the “excess” DOFs can be utilized to perform several tasks simultaneously by using the singularity-robust multiple task-priority inverse kinematics framework [8], [9], which is widely used for a variety of applications [10]-[14]. This framework has been developed for *equality tasks*, which specify exactly one desired value for certain given states of the system, for instance the position and orientation of the end effector. However, for a general robotic system, several goals may be described as *set-based tasks*, which are tasks that have a desired interval of values rather than one exact desired value. Such tasks are also referred to as inequality constraints. Examples of such tasks are staying within joint limits [15], collision/obstacle avoidance [16] and maintaining a high manipulability.

A method to incorporate set-based tasks into the singularity-robust multiple task-priority inverse kinematics framework is presented in [17], [18] and is experimentally validated in [19]. This approach entails that a set-based task is ignored while the task value is within its valid set, and the remaining tasks of the system then decide the system trajectory. When the system is on the border of the valid set, the set-based task either remains ignored, or it is implemented as an equality task with the goal of keeping the task on the boundary. The proposed algorithm will choose the latter if the other tasks of the system will push the set-based task out of its valid set. In the opposite case, the set-based task is still ignored, since meeting the requirements of the other tasks will still keep the system in the valid set, and no other action is thus needed. This results in a switched system with 2^{n_t} modes, where n_t is the number of set-based tasks.

Today, spray painting in manufacturing is mostly performed by robotic systems, and it is crucial that this task is performed both with high quality and in an efficient manner [20]. In this paper, we represent the paint trajectory as a lawn mowing pattern defined by a radius of the turns, length of the straight line segments and initial position, but the proposed method is applicable also for other paths, which can be optimized for instance in terms of speed, coverage, and paint waste [21]-[25].

The standard method of spray painting consists of applying paint while keeping the spray nozzle normal to the spray surface. In [26], for instance, the spray surface is divided into sections and the section normal, which is the weighted average of the surface normals in that section, is used to

¹S. Moe and K. Y. Pettersen are with the NTNU Center for Autonomous Marine Operations and Systems (NTNU AMOS), at Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway {signe.moe, kristin.y.pettersen}@ntnu.no

²J. T. Gravdahl is with Department of Engineering Cybernetics, NTNU, Trondheim, Norway jan.tommy.gravdahl@itk.ntnu.no

generate the paint trajectory. However, research shows that a small deviation in the orientation of the end effector relative to the surface does not affect the quality of the paint job [27], [28]. In [29], [30], for instance, path planning methods are presented to ensure that the paint thickness is within a valid interval by minimizing a cost function dependent on the angle between the spray gun axis and the surface normal. To ensure a sufficiently high paint quality, it is far more important to maintain constant velocity throughout the trajectory [31]. This is exploited in [20], where the nonlinear orientation constraints are transformed into positive definiteness constraints imposed on certain symmetric matrices to find the desired orientation at every time-step. It is shown that this approach allows the end effector to maintain a higher constant velocity throughout the trajectory guaranteeing uniform paint coating and substantially reducing the time needed to paint the object, something that is experimentally validated in [32].

In this paper, we suggest to define the angle between the paint nozzle and the spray surface as a high-priority set-based task in the control system. Thus, the entire spray process consists of one equality tasks (the spray task) in addition to the set-based orientation task, and the approach in [18] may be applied to generate reference velocities for the system joints. It has been proven that this method ensures satisfaction of the set-based tasks and asymptotic convergence of the equality task error given that the reference velocities are tracked and the tasks are compatible [33]. Unlike [20], where an optimization problem must be solved every time-step, the proposed method is deterministic and not dependent on fast convergence of numeric optimization solvers. Finally, in [20], [29], [30], the orientation of the spray nozzle is actively controlled throughout the entire operation, thereby occupying one or more DOFs at all times [20], whereas in the approach proposed in this paper, the orientation evolves freely according to the equality tasks until it is necessary to actively prevent the set-based task from being violated. Thus, the system has greater freedom to accomplish the spray task, which consists of *the pointing task* (describes the point of intersection between the central axis of the end effector and the surface, i.e. the point where the paint will hit the surface [20]) and *the distance task* (the distance between the spray nozzle and the point of intersection). To ensure uniform coating, the latter is kept constant.

This paper is organized as follows. Section II briefly describes the singularity-robust task-priority inverse kinematics framework, upon which the proposed method is built, and an introduction to set-based control is given in Section III. The proposed algorithm for a spray paint scenario is presented in Section IV, and is validated by experimental results in Section V. Finally, conclusions are given in Section VI.

II. THE SINGULARITY-ROBUST TASK-PRIORITY INVERSE KINEMATICS FRAMEWORK

This section presents the framework upon which set-based control is built.

A general robotic system has n DOFs. Its configuration is given by the joint values $\mathbf{q} = [q_1, q_2, \dots, q_n]^T$. It is then possible to express tasks and task velocities in the operational space through forward kinematics and the task Jacobian matrix. The task variable that is to be controlled is given as $\boldsymbol{\sigma}(t) \in \mathbb{R}^m$,

$$\boldsymbol{\sigma}(t) = \mathbf{f}(\mathbf{q}(t)), \quad (1)$$

where $\mathbf{f}(\mathbf{q}(t))$ is the forward kinematics, which can be derived for instance through the Denavit-Hartenberg convention [34]. The time-derivative of the task is given as

$$\dot{\boldsymbol{\sigma}}(t) = \frac{\partial \mathbf{f}(\mathbf{q}(t))}{\partial \mathbf{q}} \dot{\mathbf{q}}(t) = \mathbf{J}(\mathbf{q}(t)) \dot{\mathbf{q}}(t), \quad (2)$$

where $\mathbf{J}(\mathbf{q}(t)) \in \mathbb{R}^{m \times n}$ is the configuration-dependent analytical task Jacobian matrix and $\dot{\mathbf{q}}(t) \in \mathbb{R}^n$ is the system velocity. For compactness, the argument \mathbf{q} of tasks and Jacobians are omitted from the equations for the remainder of this section.

Consider a single m -dimensional task to be followed, with a defined desired trajectory $\boldsymbol{\sigma}_{\text{des}}(t) \in \mathbb{R}^m$. The corresponding joint references $\mathbf{q}_{\text{des}}(t) \in \mathbb{R}^n$ for the robotic system may be computed by integrating the locally inverse mapping of (2) achieved by imposing minimum-norm velocity. The following least-squares solution is given:

$$\dot{\mathbf{q}}_{\text{des}} = \mathbf{J}^\dagger \dot{\boldsymbol{\sigma}}_{\text{des}} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1} \dot{\boldsymbol{\sigma}}_{\text{des}}, \quad (3)$$

where \mathbf{J}^\dagger , implicitly defined in the above equation for full row rank matrices, is the right pseudoinverse of \mathbf{J} . In the general case, the pseudoinverse is the matrix that satisfies the four Moore-Penrose conditions (4)-(7) [35], and it is well-defined also for systems that are not square ($m \neq n$) nor have full rank [4]:

$$\mathbf{J}\mathbf{J}^\dagger \mathbf{J} = \mathbf{J}, \quad (4)$$

$$\mathbf{J}^\dagger \mathbf{J}\mathbf{J}^\dagger = \mathbf{J}^\dagger, \quad (5)$$

$$(\mathbf{J}\mathbf{J}^\dagger)^* = \mathbf{J}\mathbf{J}^\dagger, \quad (6)$$

$$(\mathbf{J}^\dagger \mathbf{J})^* = \mathbf{J}^\dagger \mathbf{J}. \quad (7)$$

Here, \mathbf{J}^* denotes the complex-conjugate of \mathbf{J} .

The vector \mathbf{q}_{des} , which is found by taking the time integral of (3), is prone to drifting. To handle this, a closed loop inverse kinematics (CLIK) version of the algorithm is usually implemented [36].

$$\dot{\mathbf{q}}_{\text{des}} = \mathbf{J}^\dagger (\dot{\boldsymbol{\sigma}}_{\text{des}} + \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}}), \quad (8)$$

where $\tilde{\boldsymbol{\sigma}} \in \mathbb{R}^m$ is the task error defined as

$$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}_{\text{des}} - \boldsymbol{\sigma} \quad (9)$$

and $\boldsymbol{\Lambda} \in \mathbb{R}^{m \times m}$ is a positive-definite matrix of gains. This feedback approach reduces the error dynamics to

$$\begin{aligned} \dot{\tilde{\boldsymbol{\sigma}}} &= \dot{\boldsymbol{\sigma}}_{\text{des}} - \dot{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}}_{\text{des}} - \mathbf{J}\dot{\mathbf{q}} \\ &= \dot{\boldsymbol{\sigma}}_{\text{des}} - \mathbf{J}\mathbf{J}^\dagger (\dot{\boldsymbol{\sigma}}_{\text{des}} + \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}}) \\ &= -\boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}}, \end{aligned} \quad (10)$$

if $\dot{\mathbf{q}} = \dot{\mathbf{q}}_{\text{des}}$ and \mathbf{J} has full rank, implying that $\mathbf{J}\mathbf{J}^\dagger = \mathbf{I}$. Equation (10) describes a linear system with a globally exponentially stable equilibrium point $\tilde{\boldsymbol{\sigma}} = \mathbf{0}$. It is worth noticing that the assumption $\dot{\mathbf{q}} = \dot{\mathbf{q}}_{\text{des}}$ is common to all inverse kinematics algorithms [9], [37]. For practical applications, it requires that the low level dynamic control loop is faster than the kinematic one.

III. SET-BASED CONTROL

This section presents the idea behind set-based control within the singularity-robust multiple task-priority inverse kinematics framework. This framework has been developed for equality tasks that have a specific desired value $\sigma_{\text{des}}(t)$, e.g. the desired end effector position. Set-based control is a method to extend the existing framework to handle set-based tasks such as the avoidance of joint limits and obstacles, field of view etc.

A set-based task is expressed through forward kinematics (1), but the objective is to keep the task in a defined set D rather than controlling it to a desired value. Mathematically, this can be expressed as $\sigma(t) \in D \forall t$ rather than $\sigma(t) = \sigma_{\text{des}}(t)$. Thus, set-based tasks cannot be directly inserted into the singularity-robust multiple task-priority inverse kinematics framework.

Consider Fig. 1. A set-based task σ is defined as *satisfied* when it is contained in its valid set, i.e. $\sigma \in D = [\sigma_{\text{min}}, \sigma_{\text{max}}]$. On the boundary of D the task is still satisfied, but it might be necessary to actively handle the task to prevent it from becoming violated.

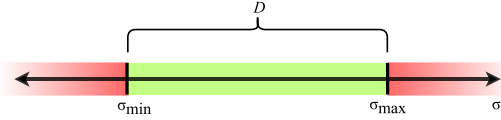


Fig. 1: Illustration of the valid set D . The set-based task σ is satisfied in D and violated outside of D .

In [17]-[33] a method is presented that allows a general number of scalar set-based tasks to be handled in this framework with a given priority within a number of equality tasks, and experimental results and practical implementation are given in [19], [33]. The results in this paper are based on these methods, and a crucial element to set-based control is the tangent cone to the set D , which is given as

$$T_D(\sigma) = \begin{cases} [0, \infty) & \sigma = \sigma_{\text{min}} \\ \mathbb{R} & \sigma \in (\sigma_{\text{min}}, \sigma_{\text{max}}) \\ (-\infty, 0] & \sigma = \sigma_{\text{max}} \end{cases} \quad (11)$$

Note that if $\dot{\sigma}(t) \in T_D(\sigma(t)) \forall t \geq t_0$, then this implies that $\sigma(t) \in D \forall t \geq t_0$. If σ is in the interior of D , the derivative is always in the tangent cone, as this is defined as \mathbb{R} . If $\sigma = \sigma_{\text{min}}$, the task is at the lower border of the set. In this case, if $\dot{\sigma} \in [0, \infty)$, then σ will either stay on the border, or move into the interior of the set. Similarly, if $\sigma = \sigma_{\text{max}}$ and $\dot{\sigma} \in (-\infty, 0]$, σ will not leave D . Hence, the goal of set-based control is to define one or more tasks with corresponding valid sets, and ensure that the task derivatives are always contained in the tangent cone of said sets. The tangent cone implementation is given in Algorithm 1 and illustrated in Figure 2.

```

Input:  $\dot{\sigma}, \sigma, \sigma_{\text{min}}, \sigma_{\text{max}}$ 
1 if  $\sigma_{\text{min}} < \sigma < \sigma_{\text{max}}$  then
2   | return True;
3 else if  $\sigma \leq \sigma_{\text{min}}$  and  $\dot{\sigma} \geq 0$  then
4   | return True;
5 else if  $\sigma \leq \sigma_{\text{min}}$  and  $\dot{\sigma} < 0$  then
6   | return False;
7 else if  $\sigma \geq \sigma_{\text{max}}$  and  $\dot{\sigma} \leq 0$  then
8   | return True;
9 else
10  | return False;
11 end

```

Algorithm 1: The boolean function in_T_C.

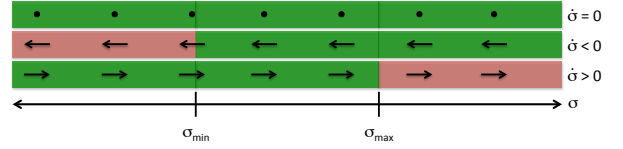


Fig. 2: Graphic illustration of the tangent cone function in_T_C with return value **True** shown in green and **False** in red

IV. SET-BASED CONTROL OF SPRAY PAINT

This section presents the proposed implementation of set-based control for a spray paint scenario. Traditionally, paint is applied by a robotic system with a nozzle that is controlled to be perpendicular to the spray surface. However, it can be shown that the velocity of the paint gun is far more important than the orientation when it comes to uniform paint coating. A small orientation deviation ($\leq 20^\circ$) of the paint gun does not affect the quality of the coating to the same extent as changes in the velocity [20]. Based on this, we choose to describe the the angle between the paint gun and the surface normal as a set-based task, and perform the spray paint operation using set-based control.

A. EXPERIMENTAL SETUP

To explore the effects of set-based control for a spray paint scenario, experiments have been run on UR5 manipulator from Universal robots. The UR5 has 6 revolute joints, and the joint angles are denoted $\mathbf{q} \triangleq [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6]^T$. The Denavit-Hartenberg (D-H) parameters are used to calculate the forward kinematics (i.e. the position and the orientation of the end effector as a function of \mathbf{q}). The parameters are given in [38] and are presented Table I with the corresponding coordinate frames illustrated in Fig. 3. The resulting forward kinematics has been experimentally verified to confirm the correctness of the parameters.

Joint	a_i [m]	α_i [rad]	d_i [m]	θ_i [rad]
1	0	$\pi/2$	0.089	q_1
2	-0.425	0	0	q_2
3	-0.392	0	0	q_3
4	0	$\pi/2$	0.109	q_4
5	0	$-\pi/2$	0.095	q_5
6	0	0	0.082	q_6

TABLE I: Table of the D-H parameters of the UR5. The corresponding coordinate systems can be seen in Fig. 3.

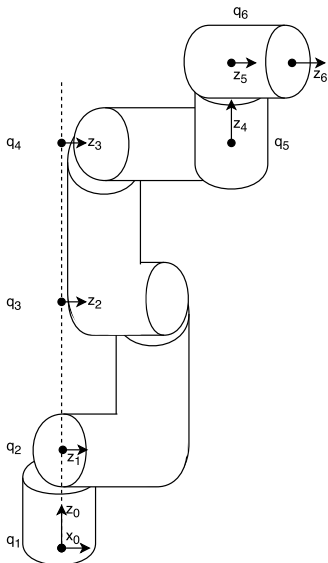


Fig. 3: Coordinate frames corresponding to the D-H parameters in Table I.

The UR5 is equipped with a high-level controller that can control the robot both in joint and Cartesian space. In the experiments presented here, a calculated reference \mathbf{q}_{des} is sent to the high-level controller, which is assumed to function nominally such that

$$\mathbf{q} \approx \mathbf{q}_{\text{des}}. \quad (12)$$

From this reference, $\dot{\mathbf{q}}_{\text{des}}$ and $\ddot{\mathbf{q}}_{\text{des}}$ are extrapolated and sent with \mathbf{q}_{des} to the low-level controller.

The structure of the system is illustrated in Fig. 4. The algorithm described in Section IV-C is implemented in the kinematic controller block. Every timestep, a reference for the joint velocities is calculated and integrated to obtain the desired joint angles \mathbf{q}_{des} . This is used as input to the dynamic controller, which in turn applies torques to the joint motors. Note that the actual state \mathbf{q} is not used for feedback to the kinematic control block. When the current state is used as input for the kinematic controller, the kinematic and dynamic loops are coupled and the gains designed for the kinematic control alone according to [33] can not be used. This would result in uneven motion, and therefore the kinematic control block instead receives the previous reference as feedback, which leads to much nicer behavior, and is a good approximation because the dynamic controller tracks the reference with very high precision. This is the standard method of implementation for industry robots when kinematic control is used.

The communication between the implemented algorithm and the industrial manipulator system occurs through a TCP/IP connection which operates at 125 Hz. The kinematic control block is implemented using python, which is a very suitable programming language for the task. In particular, the TCP/IP connection is very simple to set up in Python. Furthermore, python has several libraries that can handle different math and matrix operations.

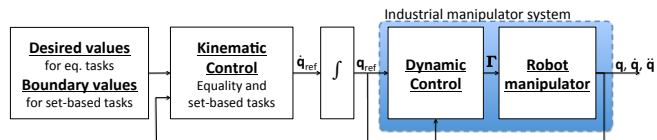


Fig. 4: The control structure of the experiments. The tested algorithm is implemented in the kinematic controller block.

B. IMPLEMENTED TASKS

Several different tasks make up the basis for a spray paint scenario.

1) *Position control*: The position of the end effector $\mathbf{p}_e = [x_e \ y_e \ z_e]^T$ (see Fig. 5) relative to the base coordinate frame is given by the forward kinematics. The analytical expression can be found through the homogeneous transformation matrix [34] using the D-H parameters given in Table I. The task is then defined by

$$\boldsymbol{\sigma}_{\text{pos}}(\mathbf{q}) = \mathbf{p}_e(\mathbf{q}) = \mathbf{f}(\mathbf{q}) \in \mathbb{R}^3 \quad (13)$$

$$\dot{\boldsymbol{\sigma}}_{\text{pos}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_{\text{pos}}(\mathbf{q})\dot{\mathbf{q}} = \frac{d\mathbf{f}}{d\mathbf{q}}\dot{\mathbf{q}}, \quad (14)$$

where the function $\mathbf{f}(\mathbf{q})$ is given by the forward kinematics.

2) *Field of view*: The field of view is defined as the outgoing unit vector of the end effector, i.e. the z_6 -axis in Figure 3. This vector expressed in base coordinates is denoted $\mathbf{a}(\mathbf{q}) = [a_x(\mathbf{q}) \ a_y(\mathbf{q}) \ a_z(\mathbf{q})]^T \in \mathbb{R}^3$, and can be found through the homogeneous transformation matrix using the D-H parameters.

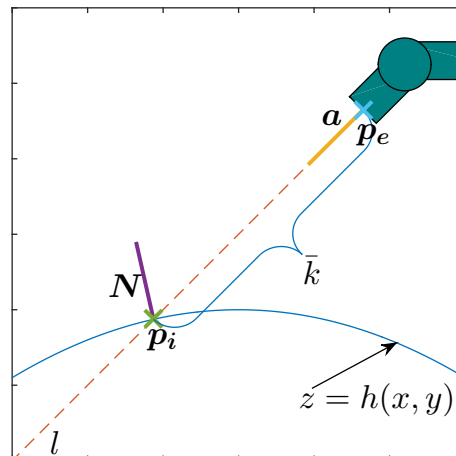


Fig. 5: Illustration of the spray paint scenario.

It is very useful to control the FOV when directional devices, sensors or, as in this particular case, a spray nozzle are mounted on the end-effector and these are desired to point in a certain direction $\mathbf{a}_{\text{des}}(\mathbf{q}) \in \mathbb{R}^3$. We then define a one-dimensional FOV task as the norm of the error between \mathbf{a} and \mathbf{a}_{des} :

$$\sigma_{\text{FOV}}(\mathbf{q}) = \sqrt{(\mathbf{a}_{\text{des}}(\mathbf{q}) - \mathbf{a}(\mathbf{q}))^T (\mathbf{a}_{\text{des}}(\mathbf{q}) - \mathbf{a}(\mathbf{q}))} \in \mathbb{R} \quad (15)$$

$$\begin{aligned} \dot{\sigma}_{\text{FOV}}(\mathbf{q}, \dot{\mathbf{q}}) &= \mathbf{J}_{\text{FOV}}(\mathbf{q}) \dot{\mathbf{q}} \\ &= \frac{(\mathbf{a}_{\text{des}}(\mathbf{q}) - \mathbf{a}(\mathbf{q}))^T}{\sigma_{\text{FOV}}(\mathbf{q})} \left(\frac{d\mathbf{a}_{\text{des}}(\mathbf{q})}{d\mathbf{q}} - \frac{d\mathbf{a}(\mathbf{q})}{d\mathbf{q}} \right) \dot{\mathbf{q}} \end{aligned} \quad (16)$$

Note that \mathbf{J}_{FOV} is not defined for $\sigma_{\text{FOV}} = 0$. In the implementation, this is solved by adding a small $\varepsilon > 0$ to the denominator of this Jacobian, thereby ensuring that division by zero does not occur.

In these experiments, the vector \mathbf{a} corresponds to the direction of the spray nozzle and is therefore highly relevant for a spray paint scenario. For a traditional spray paint scenario, $\mathbf{a}_{\text{des}} = -\mathbf{N}$, where \mathbf{N} is the normal to the spray surface. This is illustrated in Fig. 5.

3) *Spray task*: For spray painting, it is desirable to control the point of intersection $\mathbf{p}_i \in \mathbb{R}^2$, which is the point where the spray hits the spray surface. This surface is described by the function $h(x, y)$ such that

$$z = h(x, y), \quad (17)$$

and is illustrated in Fig. 5. The intersection point is given as the point where the line l intersects the surface $z = h(x, y)$. The line l can be parametrized by $k \geq 0$ as

$$l(k, \mathbf{q}) := \mathbf{p}_e(\mathbf{q}) + \mathbf{a}(\mathbf{q})k = \begin{bmatrix} x_e(\mathbf{q}) + a_x(\mathbf{q})k \\ y_e(\mathbf{q}) + a_y(\mathbf{q})k \\ z_e(\mathbf{q}) + a_z(\mathbf{q})k \end{bmatrix} \quad (18)$$

The point of intersection occurs for $k = \bar{k}(\mathbf{q})$, which can be calculated by solving the equation

$$z_e(\mathbf{q}) + a_z(\mathbf{q})\bar{k} = h(x_e(\mathbf{q}) + a_x(\mathbf{q})\bar{k}, y_e(\mathbf{q}) + a_y(\mathbf{q})\bar{k}). \quad (19)$$

Note that depending on the surface function h , the above equation may not have an analytical solution for \bar{k} . For practical purposes \bar{k} may be estimated by using optimization techniques.

In addition to controlling the point of intersection, it is necessary to control the distance between the surface and the spray gun, which is given by \bar{k} , to achieve a uniform paint coat. Since it may be challenging to find an explicit expression for \bar{k} , the task Jacobian is calculated by approximating the spray surface as the tangent plane in the intersection point. Hence, for implementation purposes, the following steps are taken: First, the numeric value for \bar{k} and the resulting \mathbf{p}_i are calculated based on (19) using optimization techniques, and

$$\mathbf{p}_i(\mathbf{q}) = \begin{bmatrix} x_i(\mathbf{q}) \\ y_i(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} x_e(\mathbf{q}) \\ y_e(\mathbf{q}) \end{bmatrix} + \bar{k}(\mathbf{q}) \begin{bmatrix} a_x(\mathbf{q}) \\ a_y(\mathbf{q}) \end{bmatrix}, \quad (20)$$

$$z_i(\mathbf{q}) = h(x_i(\mathbf{q}), y_i(\mathbf{q})). \quad (21)$$

Then, the normal vector \mathbf{N} to the surface $z = h(x, y)$ in the

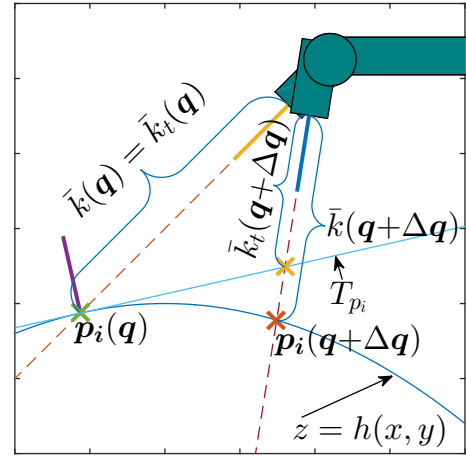


Fig. 6: Illustration of \bar{k} and \bar{k}_t .

point (\mathbf{p}_i, z_i) is found as

$$\mathbf{N}(\mathbf{q}) = \begin{bmatrix} -\frac{\partial h}{\partial x}(x_i(\mathbf{q}), y_i(\mathbf{q})) \\ -\frac{\partial h}{\partial y}(x_i(\mathbf{q}), y_i(\mathbf{q})) \\ 1 \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}, \quad (22)$$

and the tangent plane in the same point is given as

$$T_{\mathbf{p}_i} : n_1(x - x_i) + n_2(y - y_i) + n_3(z - z_i) = 0. \quad (23)$$

Note that even though the point \mathbf{p}_i and \mathbf{N} change as a function of the robot configuration \mathbf{q} , the above equation describes the tangent plane in one specific \mathbf{p}_i , and thus these values are constants. Although it may be challenging, or even impossible, to derive an explicit expression for the distance \bar{k} between the end effector position \mathbf{p}_e and a general surface $z = h(x, y)$ along the line l , it is straight forward to calculate the same distance between the end effector position and the tangent plane. This distance is denoted \bar{k}_t . To do so, we insert (18) into (23), which yields

$$\begin{aligned} \bar{k}_t(\mathbf{q}) &= -\frac{n_1(x_e(\mathbf{q}) - x_i) + n_2(y_e(\mathbf{q}) - y_i) + n_3(z_e(\mathbf{q}) - z_i)}{n_1 a_x(\mathbf{q}) + n_2 a_y(\mathbf{q}) + n_3 a_z(\mathbf{q})} \\ &= -\frac{a(\mathbf{q})}{b(\mathbf{q})}. \end{aligned} \quad (24)$$

Then, taking the time derivative of (24), it is clear that

$$\begin{aligned} \dot{\bar{k}}_t(\mathbf{q}, \dot{\mathbf{q}}) &= -\frac{(n_1 \dot{x}_e(\mathbf{q}, \dot{\mathbf{q}}) + n_2 \dot{y}_e(\mathbf{q}, \dot{\mathbf{q}}) + n_3 \dot{z}_e(\mathbf{q}, \dot{\mathbf{q}}))b(\mathbf{q})}{b(\mathbf{q})^2} \\ &\quad + \frac{a(\mathbf{q})(n_1 \dot{a}_x(\mathbf{q}, \dot{\mathbf{q}}) + n_2 \dot{a}_y(\mathbf{q}, \dot{\mathbf{q}}) + n_3 \dot{a}_z(\mathbf{q}, \dot{\mathbf{q}}))}{b(\mathbf{q})^2} \\ &= -\frac{b(\mathbf{q})(n_1 \mathbf{J}_{\text{pos}_1}(\mathbf{q}) + n_2 \mathbf{J}_{\text{pos}_2}(\mathbf{q}) + n_3 \mathbf{J}_{\text{pos}_3}(\mathbf{q}))}{b(\mathbf{q})^2} \dot{\mathbf{q}} \\ &\quad + \frac{a(\mathbf{q})(n_1 \mathbf{J}_{\text{FOV}_1}(\mathbf{q}) + n_2 \mathbf{J}_{\text{FOV}_2}(\mathbf{q}) + n_3 \mathbf{J}_{\text{FOV}_3}(\mathbf{q}))}{b(\mathbf{q})^2} \dot{\mathbf{q}} \\ &= \mathbf{J}_{\text{dist}_t}(\mathbf{q}) \dot{\mathbf{q}}, \end{aligned} \quad (25)$$

where \mathbf{J}_{A_i} denotes the i th row of the matrix \mathbf{J}_A . For small changes $\dot{\mathbf{q}}$ (i.e. small time steps), the approximation

$$\dot{\bar{k}} \approx \dot{\bar{k}}_t = \mathbf{J}_{\text{dist}_t}(\mathbf{q}) \dot{\mathbf{q}} \quad (26)$$

is sufficient. Thus, in the implementation, we estimate the corresponding change in \bar{k} related to a change in configuration $\dot{\mathbf{q}}$ as the change in the distance between the end effector and the tangent plane $T_{\mathbf{p}_i}$ (23), see Fig. 6. Note that this

approach does not result in drifting over time, as the exact \bar{k} is found numerically and a new tangent plane is calculated at each time step.

Hence, we define the spray task $\sigma_{\text{spray}} \in \mathbb{R}^3$ as

$$\sigma_{\text{spray}}(\mathbf{q}) = \begin{bmatrix} \mathbf{p}_i(\mathbf{q}) \\ \bar{k}(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} x_e(\mathbf{q}) + a_x(\mathbf{q})\bar{k}(\mathbf{q}) \\ y_e(\mathbf{q}) + a_y(\mathbf{q})\bar{k}(\mathbf{q}) \\ \bar{k}(\mathbf{q}) \end{bmatrix}, \quad (27)$$

$$\begin{aligned} \dot{\sigma}_{\text{spray}}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{bmatrix} \dot{x}_e(\mathbf{q}, \dot{\mathbf{q}}) + \dot{a}_x(\mathbf{q}, \dot{\mathbf{q}})\bar{k}(\mathbf{q}) + a_x(\mathbf{q})\dot{\bar{k}}(\mathbf{q}, \dot{\mathbf{q}}) \\ \dot{y}_e(\mathbf{q}, \dot{\mathbf{q}}) + \dot{a}_y(\mathbf{q}, \dot{\mathbf{q}})\bar{k}(\mathbf{q}) + a_y(\mathbf{q})\dot{\bar{k}}(\mathbf{q}, \dot{\mathbf{q}}) \\ \dot{\bar{k}}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} \\ &\approx \underbrace{\begin{bmatrix} \mathbf{J}_{\text{pos}_1}(\mathbf{q}) + \bar{k}(\mathbf{q})\mathbf{J}_{\text{FOV}_1}(\mathbf{q}) + a_x(\mathbf{q})\mathbf{J}_{\text{dist}_t}(\mathbf{q}) \\ \mathbf{J}_{\text{pos}_2}(\mathbf{q}) + \bar{k}(\mathbf{q})\mathbf{J}_{\text{FOV}_2}(\mathbf{q}) + a_y(\mathbf{q})\mathbf{J}_{\text{dist}_t}(\mathbf{q}) \\ \mathbf{J}_{\text{dist}_t}(\mathbf{q}) \end{bmatrix}}_{\mathbf{J}_{\text{spray}}(\mathbf{q})} \dot{\mathbf{q}}. \end{aligned} \quad (28)$$

C. IMPLEMENTATION

In these experiments, the lawn mowing spray pattern is defined by a length L , a radius r and an initial point (x_0, y_0) , and is illustrated in Fig. 7. The pattern is parametrized by the arc length s :

$$x_{\text{spray}}(s) = \begin{cases} s + x_0, & 0 \leq s \leq L \\ L + r \sin\left(\frac{s-L}{r}\right) + x_0, & L < s \leq L + \pi r \\ L - (s - L - \pi r) + x_0, & L + \pi r < s \leq 2L + \pi r \\ -r \sin\left(\frac{s-2L-\pi r}{r}\right) + x_0, & 2L + \pi r < s \leq 2(L + \pi r), \end{cases} \quad (29)$$

$$y_{\text{spray}}(s) = \begin{cases} y_0, & 0 \leq s \leq L \\ r(-\cos\left(\frac{s-L}{r}\right) + 1) + y_0, & L < s \leq L + \pi r \\ 2r + y_0, & L + \pi r < s \leq 2L + \pi r \\ 2r + r(\cos\left(\frac{s-2L-\pi r}{r}\right) - 1) + y_0, & 2L + \pi r < s \leq 2(L + \pi r), \end{cases} \quad (30)$$

↓

$$\dot{x}_{\text{spray}}(s, \dot{s}) = \begin{cases} \dot{s}, & 0 \leq s \leq L \\ \dot{s} \cos\left(\frac{s-L}{r}\right), & L < s \leq L + \pi r \\ -\dot{s}, & L + \pi r < s \leq 2L + \pi r \\ -\dot{s} \cos\left(\frac{s-2L-\pi r}{r}\right), & 2L + \pi r < s \leq 2(L + \pi r), \end{cases} \quad (31)$$

$$\dot{y}_{\text{spray}}(s, \dot{s}) = \begin{cases} 0, & 0 \leq s \leq L \\ \dot{s} \sin\left(\frac{s-L}{r}\right), & L < s \leq L + \pi r \\ 0, & L + \pi r < s \leq 2L + \pi r \\ -\dot{s} \sin\left(\frac{s-2L-\pi r}{r}\right), & 2L + \pi r < s \leq 2(L + \pi r). \end{cases} \quad (32)$$

Note that the arc length s is a virtual parameter that we are free to choose. In these experiments, we have chosen

$$s(t) = Ut, \quad (33)$$

where $U > 0$ is a constant and represents the velocity with which the spray is applied along the surface. Therefore, the spray pattern is in effect a function of time, $x_{\text{spray}}(s(t))$ and $y_{\text{spray}}(s(t))$. Note that any spray pattern given as a function of time may be applied for the method proposed in this paper, and that the typical spray pattern (29)-(32) is chosen as an example. Also note that $\dot{s}(t) = U$, and that U is the velocity of the trajectory at the paint surface and not of the end effector itself.

Furthermore, the surface is described by the function $z = h(x, y)$. For instance, a flat surface on the xy -plane would yield $h(x, y) = c$ for some constant c , whereas the curved

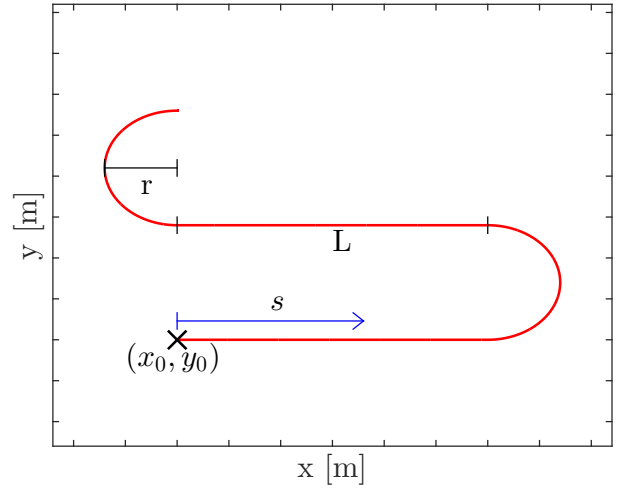


Fig. 7: Spray pattern fully defined by the length L , radius r and initial point (x_0, y_0) .

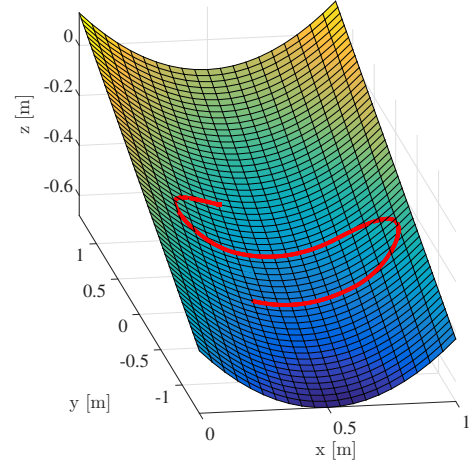


Fig. 8: Example of spray surface and spray pattern.

surface given by

$$z = h(x, y) = (x - 0.5)^2 + 0.2y - 0.4 \quad (34)$$

and the resulting spray pattern for $L = 0.4$ m, $r = 0.2$ m, $x_0 = 0.35$ m and $y_0 = -0.55$ m are illustrated in Fig. 8 as an example. The spray pattern is given in spatial coordinates by

$$\mathbf{v}(s) = \begin{bmatrix} x_{\text{spray}}(s) \\ y_{\text{spray}}(s) \\ h(x_{\text{spray}}(s), y_{\text{spray}}(s)) \end{bmatrix}. \quad (35)$$

Having established the spray task, we use FOV as a set-based task to ensure that the angle between the spray direction and the surface normal does not exceed a maximum limit of θ . Hence, according to set-based theory [33], the resulting system has two modes, one where the orientation evolves freely, and one where the FOV error is frozen at the allowed maximum limit. For the former, we denote the system task as

$$\sigma_1(\mathbf{q}) = \sigma_{\text{spray}}(\mathbf{q}), \quad (36)$$

which is defined in (27) and

$$\boldsymbol{\sigma}_{1,\text{des}}(t) = \begin{bmatrix} x_{\text{spray}}(s(t)) \\ y_{\text{spray}}(s(t)) \\ \bar{k}_{\text{des}} \end{bmatrix}, \quad \dot{\boldsymbol{\sigma}}_{1,\text{des}}(t) = \begin{bmatrix} \dot{x}_{\text{spray}}(s(t), \dot{s}(t)) \\ \dot{y}_{\text{spray}}(s(t), \dot{s}(t)) \\ 0 \end{bmatrix} \quad (37)$$

where $\bar{k}_{\text{des}} > 0$ is a constant desired distance between the end effector and the spray surface along the line l , and $x_{\text{spray}}(s(t))$, $y_{\text{spray}}(s(t))$ and their respective time derivatives are given in (29)-(32) with $s(t) = Ut$ and $\dot{s}(t) = U$.

Hence, we define mode 1 as

$$\mathbf{f}_1(t, \mathbf{q}) = \mathbf{J}_{\text{spray}}^\dagger(\mathbf{q})(\dot{\boldsymbol{\sigma}}_{1,\text{des}}(t) + \boldsymbol{\Lambda}_1 \tilde{\boldsymbol{\sigma}}_1(t, \mathbf{q})), \quad (38)$$

where $\boldsymbol{\Lambda}_1 > 0$ is a positive control gain matrix and $\tilde{\boldsymbol{\sigma}}_1 = \boldsymbol{\sigma}_{1,\text{des}} - \boldsymbol{\sigma}_1$ is the task error. Note that $\boldsymbol{\Lambda}_1$ corresponds to the proportional gain of a P-controller. Here, we control the spray trajectory on the surface (the point where the spray hits the surface) and the distance between the nozzle and the surface. In this mode there is no limitation on the orientation, which could exceed the maximum angle between the surface normal and spray direction, above which the quality of the paint job may deteriorate. Hence, we allow $\dot{\mathbf{q}}$ to follow the vector field \mathbf{f}_1 as long as this mode will not result in the set-based task being violated, and switch to mode 2 otherwise. Switching is determined by the tangent cone (11). In mode 2, the FOV-vector is frozen on this limit. Hence, we define

$$\boldsymbol{\sigma}_2(\mathbf{q}) = \begin{bmatrix} \boldsymbol{\sigma}_1(\mathbf{q}) \\ \boldsymbol{\sigma}_{\text{FOV}}(\mathbf{q}) \end{bmatrix}, \quad (39)$$

where $\boldsymbol{\sigma}_{\text{FOV}}(\mathbf{q})$ is defined in (15) with

$$\mathbf{a}_{\text{des}}(\mathbf{q}) = -\frac{N(\mathbf{q})}{|N(\mathbf{q})|}, \quad (40)$$

$$\boldsymbol{\sigma}_{2,\text{des}}(t) = \begin{bmatrix} \boldsymbol{\sigma}_{1,\text{des}}(t) \\ \sqrt{2(1 - \cos(\theta))} \end{bmatrix}, \quad \dot{\boldsymbol{\sigma}}_{2,\text{des}}(t) = \begin{bmatrix} \dot{\boldsymbol{\sigma}}_{1,\text{des}}(t) \\ 0 \end{bmatrix}, \quad (41)$$

where θ is the maximum allowed degrees between the FOV vector and the surface normal. Hence, we define mode 2 as

$$\mathbf{f}_2(t, \mathbf{q}) = \begin{bmatrix} \mathbf{J}_{\text{spray}}(\mathbf{q}) \\ \mathbf{J}_{\text{FOV}}(\mathbf{q}) \end{bmatrix}^\dagger (\dot{\boldsymbol{\sigma}}_{2,\text{des}} + \boldsymbol{\Lambda}_2 \tilde{\boldsymbol{\sigma}}_2). \quad (42)$$

The step-by-step implementation for an iteration in the control system is the given below:

V. EXPERIMENTAL RESULTS

In the experiments, the following parameters were used:

$$\theta = 20^\circ \quad (43)$$

$$h(x, y) = -0.45 \quad (44)$$

$$\bar{k}_{\text{des}} = 0.3 \text{ m} \quad (45)$$

$$\boldsymbol{\Lambda}_1 = 0.4 \mathbf{I}_{3 \times 3} \quad (46)$$

$$\boldsymbol{\Lambda}_2 = 0.4 \mathbf{I}_{4 \times 4} \quad (47)$$

The choice of $\theta = 20^\circ$ is the same as in [32]. Since the function $h(x, y)$ is a constant, the spray surface is flat, and the desired distance between the nozzle and the point of intersection \mathbf{p}_1 is 0.3 m. Note that in these experiments, no paint was actually applied to a surface. The algorithm controls the movement of the manipulator in such a way that it corresponds to a paint scenario. For the initial experiments we chose a flat surface to better illustrate the method, but it is in fact general and may be applied to any differentiable

Input: Time t , current configuration \mathbf{q}

- 1 Update the variable $s = U(t - t_0)$;
- 2 Calculate $\boldsymbol{\sigma}_{1,\text{des}}(t)$, $\boldsymbol{\sigma}_{2,\text{des}}(t)$, $\dot{\boldsymbol{\sigma}}_{1,\text{des}}(t)$ and $\dot{\boldsymbol{\sigma}}_{2,\text{des}}(t)$, eq. (37) and (41);
- 3 Use the forward kinematics to find $\mathbf{p}_e(\mathbf{q})$ and $\mathbf{a}(\mathbf{q})$;
- 4 Use $\mathbf{p}_e(\mathbf{q})$, $\mathbf{a}(\mathbf{q})$ and optimization techniques to calculate $\bar{k}(\mathbf{q})$, eq. (19);
- 5 Calculate $\boldsymbol{\sigma}_1(\mathbf{q})$, $\boldsymbol{\sigma}_2(\mathbf{q})$, $\mathbf{J}_{\text{spray}}(\mathbf{q})$ and $\mathbf{J}_{\text{FOV}}(\mathbf{q})$, eq (36), (39), (28) and (16);
- 6 Find $\mathbf{f}_1(t, \mathbf{q})$ and $\mathbf{f}_2(t, \mathbf{q})$, eq. (38) and (42);
- 7 Use the tangent cone function in Alg. 1 to decide mode:

$$a = \text{in_T_C}(\mathbf{J}_{\text{FOV}} \mathbf{f}_1, \boldsymbol{\sigma}_{\text{FOV}}, 0, \sqrt{2(1 - \cos(\theta))})$$

if a is *True* **then**

$$8 \quad | \quad \dot{\mathbf{q}}_{\text{des}} = \mathbf{f}_1;$$

9 else

$$10 \quad | \quad \dot{\mathbf{q}}_{\text{des}} = \mathbf{f}_2;$$

11 end

Algorithm 2: One iteration in the control system.

surface $h(x, y)$ as described in Section IV-C. Also note that although this paper assumes such a surface, the principle of using set-based theory to allow a small angle between the surface normal and spray direction is highly generic and may be adapted to other types of spray surfaces as well.

Experiments were run for three different patterns and three spray velocities:

$$(r, L) = \{(0.07, 0.3), (0.12, 0.2), (0.16, 0.1)\} \text{ m} \quad (48)$$

$$U = \{0.15, 0.10, 0.05\} \text{ m/s} \quad (49)$$

Furthermore, several set-based approaches were tested and compared to the current industry standard, in which the spray gun is kept perpendicular to the spray surface at all times. This corresponds to $\dot{\mathbf{q}}_{\text{des}} \equiv \mathbf{f}_2$, with $\theta = 0$, and is referred to as ST in Table II and III. The following set-based approaches were implemented:

- A - Pure set-based control, as described in the step-by-step implementation above
- B - Switch between ST on the straight segments and approach A in the turns
- C - Corresponds to A with smooth switching, i.e. when switching between mode 1 and 2
- D - Corresponds to B with smooth switching

To illustrate, the results are plotted in 3D with the current standard solution ST in Fig. 9 and the set-based approach A in Fig. 10, for $r = 0.16$ m, $L = 0.10$ m and $U = 0.10$ m/s.

When switching between modes in approach A and B, the reference $\dot{\mathbf{q}}_{\text{des}}$ changes abruptly, thereby requiring large joint accelerations. This could potentially be very demanding for the manipulator to handle. Hence, in approach C and D switches are detected between mode 1 and 2 and between ST, mode 1 and mode 2, respectively. The system reference velocity $\dot{\mathbf{q}}_{\text{des}}$ is then smoothed between the previous and the new reference using the following smoothing function:

$$\alpha(t, t_{\text{last_switch}}) = \frac{1}{\pi} \arctan(a(t - t_{\text{last_switch}} - b)) + \frac{1}{2} \quad (50)$$

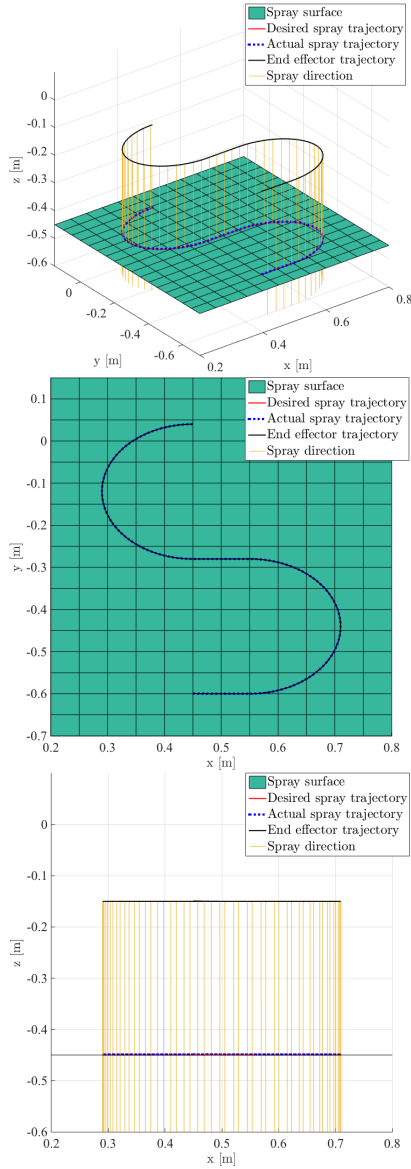


Fig. 9: Current standard solution for spray paint. The paint is applied orthogonally to the spray surface. For a flat surface, the end effector movement copies the pattern at a constant distance.

Here, the parameters a and b determine the sharpness of smoothing function and the delay in time before the transition takes place, respectively. This is illustrated in Fig. 11. In these experiments, $a = 110$ and $b = 0.05$.

Note that when smooth switching is used, the strict priority of the modes are lost during transitions. In mode 1, the FOV task evolves freely, and in mode 2 the task is kept stationary at the maximum limit of θ . Hence, if mode 2 is activated at the maximum limit and smooth switching is applied to the transition, this limit might be exceeded because mode 1 affects the solution \dot{q}_{des} during the transition. To avoid exceeding the limit, step 7 of the implementation in Alg. 2 is slightly different for approaches C and D. In this case, the

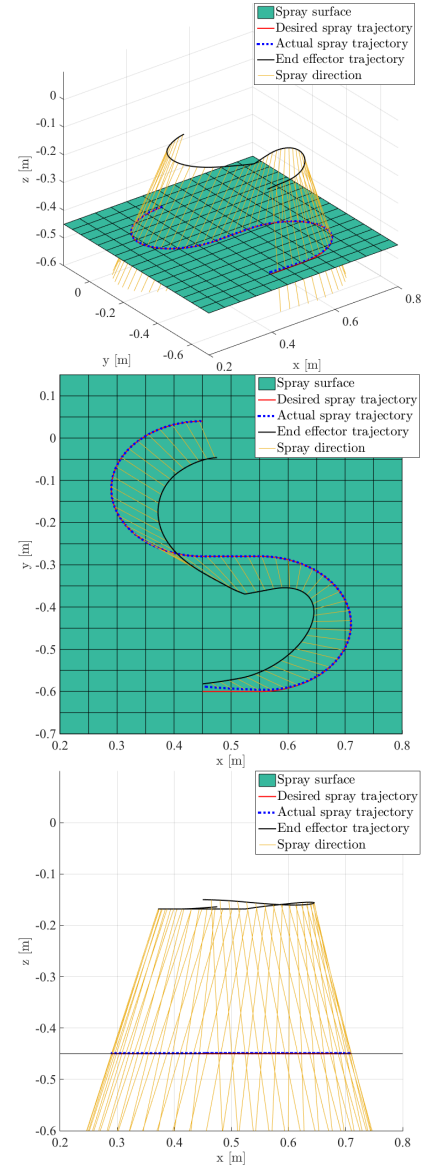


Fig. 10: Set-based solution for spray paint, approach A. The maximum allowed angle between the spray direction and the surface normal is limited to $\theta = 20^\circ$. The orientation evolves freely inside this set.

boolean variable a is defined as

$$a = \text{in_T_C}(\mathbf{J}_{\text{FOV}} \mathbf{f}_1, \sigma_{\text{FOV}}, 0, \sqrt{2(1 - \cos(\theta - \theta_0))}), \quad (51)$$

where θ_0 has been chosen as 5° . For practical purposes, this means that if the FOV angle between the spray direction and spray surface exceeds 15° , mode 2 is activated and this angle is slowly controlled towards the maximum allowed limit of $\theta = 20^\circ$. This approach is conservative, but by defining such a buffer zone, we ensure that the maximum limit is not exceeded when smooth switching is applied.

To calculate the energy consumption, we measure the total current in all the joints, see Fig. 12 and 13. The UR5 runs

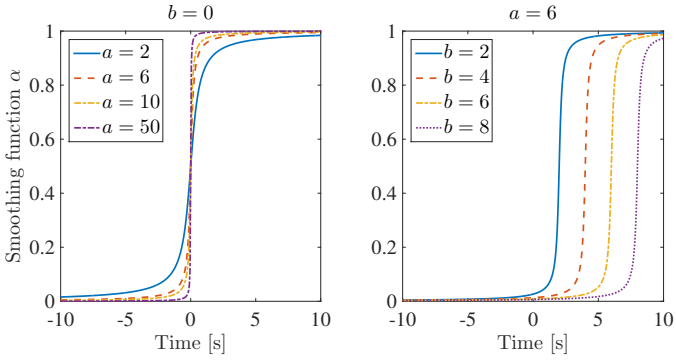


Fig. 11: Smoothing function illustrated for different parameters and $t_{\text{last_switch}} = 0$. The parameter a determines the sharpness, i.e. how long the transition takes, and b the temporal shift, i.e. how long after the last switch the smoothing function is activated.

on a 24 V supply supply¹, and the mean power is therefore given as

$$\bar{P} = 24\bar{I}_{\text{tot}}, \quad (52)$$

where \bar{I}_{tot} is the mean of the current drawn in all the joints. The total energy consumption is then given as

$$E = \bar{P}\Delta t, \quad (53)$$

where Δt is the time of the experiment.

The experimental results are shown in Table II and III. In Table II, the standard approach is compared to the set-based approaches A-D for the same spray trajectory velocity U . In all experiments, the spray pattern was repeated twice, and as the Table II shows, for the standard solution the length of the end effector path is equal to the spray pattern length $2(2\pi r + 2L)$, and the average end effector velocity is equal to the spray trajectory velocity U . This is expected, since the end effector in this approach simply copies the spray pattern while keeping the angle between the spray direction and surface equal to zero (see Fig. 9). However, for the set-based approaches, by allowing an angle between the spray direction and the surface normal, the end effector moves a shorter distance while maintaining the same spray trajectory velocity U at the spray surface. Therefore, the average velocity of the end effector is lower than U . The set-based approaches A-D thus consume less energy than the standard method ST of controlling the orientation directly, on average 4.32% less, and at best, 8.05%. Since the spray trajectory velocity U is the same for the set-based and standard solutions, the approaches spend the same amount of time to complete the spray pattern. Furthermore, the maximum angle $\theta = 20^\circ$ between the spray direction and the surface normal is never exceeded, and the desired trajectory is tracked with satisfactory accuracy. For all experiments, the root-mean-square (rms) error of the spray task (27) is 3×10^{-3} m, ranging from 5×10^{-4} m to 1.4×10^{-2} m. The rms-error depends on the velocity U and the chosen approach. In

particular, larger values for U result in a slightly larger rms-error. A possible solution to reduce the rms-error is to increase the task gains Λ_1 and Λ_2 . Furthermore, experiments show that the rms-errors for the set-based approaches C and D are slightly larger than the other solutions. As mentioned above, during smooth switching the strict priority of the tasks are lost, which affects the task errors. To minimize this effect, the smoothing function should be as aggressive as possible, i.e. should have a large value for a in (50).

Maintaining a constant, high speed during turns may be challenging for a robotic system because of the large torques required. In the case that the specifications of the robot is the limiting factor in the spray process a set-based approach may increase the velocity, since the end effector in this case is not required to copy the pattern directly, but can use the freedom in orientation to achieve the same spray result with a less demanding trajectory for the end effector. To investigate the consequences of increasing the end effector speed, the experiments from Table II were repeated, but in this case U was adjusted so that the average velocity of the end effector was the same in all experiments. The results are shown in Table III.

By increasing the spray trajectory velocity U such that the resulting average end effector velocity was the same for all approaches, the set-based approaches completed the spray task in less time. On average, the approaches A-D spent 74.50% of the time compared to the standard solution, and at best 66.73%. The energy consumption was also significantly reduced, and the set-based approaches in this case consumed on average 27.46%, and at best 34.98%, less than the standard approach. The main reason for this is the reduced operation time, but as observed in Table II, the set-based approaches are also more energy efficient than the standard solution in general.

In addition, we can conclude that the pure set-based approach A is more energy and time efficient than approach B, which can be seen as a hybrid between set-based (active on the turns) and the standard solution (active on the straight lines). This is also reflected in the smooth switching versions, where C is more energy and time efficient than D. Future work will therefore be based on approach A and C.

When comparing abrupt and smooth switching, Table II and III show that the smooth approaches C and D perform similarly to their respective original versions A and B. However, the smooth switching ensures that the commanded joint accelerations are not too high, making these approaches feasible also for higher velocities that would result in a security stop for the abrupt methods A and B. Approach C and D also consume less energy than A and B when the velocities are large since the abrupt switches require a large amount of energy compared to the more conservative, yet smooth methods. This is confirmed by Fig. 12 and 13, where the active mode, set-based task $\sigma_{\text{FOV}}(\mathbf{q})$, total current in all joints and joint velocities are plotted for approach A and C with $U = 0.10$ m/s, $r = 16$ m, $L = 0.1$ m. In mode 1, the set-based task evolves freely according to the spray task, and in mode 2 it is controlled to the maximum limit

¹UR5 user manual: https://www.universal-robots.com/media/8704/ur5_user_manual_gb.pdf

		U = 0.15 m/s					U = 0.10 m/s					U = 0.05 m/s				
		ST	A	B	C	D	ST	A	B	C	D	ST	A	B	C	D
r = 0.07 m L = 0.3 m	Time [s]	13,87	13,87	13,87	13,86	13,86	20,80	20,80	20,80	20,80	20,80	41,60	41,60	41,60	41,60	41,60
	Ee path [m]	2,08	1,39	1,69	1,43	1,75	2,08	1,39	1,73	1,42	1,78	2,08	1,39	1,79	1,41	1,82
	Avg. ee vel. [m/s]	0,15	0,10	0,12	0,10	0,13	0,10	0,07	0,08	0,07	0,09	0,05	0,03	0,04	0,03	0,04
	Energy [%]	100	95,67	98,70	95,27	97,94	100	97,35	99,99	96,63	97,93	100	97,11	98,99	97,30	99,86
r = 0.12 m L = 0.2 m	Time [s]	15,39	15,39	15,39	15,39	15,39	23,08	23,09	23,09	23,08	23,08	46,17	46,17	46,16	46,16	46,16
	Ee path [m]	2,31	1,63	1,70	1,62	1,73	2,31	1,62	1,69	1,62	1,74	2,31	1,62	1,78	1,62	1,81
	Avg. ee vel. [m/s]	0,15	0,11	0,11	0,11	0,11	0,10	0,07	0,07	0,07	0,08	0,05	0,04	0,04	0,04	0,04
	Energy [%]	100	93,25	96,99	92,62	95,37	100	92,50	94,82	91,95	93,71	100	94,83	96,77	94,17	97,34
r = 0.16 m L = 0.1 m	Time [s]	16,07	16,07	16,07	16,07	16,07	24,11	24,11	24,11	24,11	24,11	48,22	48,22	48,22	48,21	48,21
	Ee path [m]	2,42	1,78	1,82	1,77	1,83	2,42	1,78	1,82	1,78	1,83	2,42	1,78	1,81	1,77	1,83
	Avg. ee vel. [m/s]	0,15	0,11	0,11	0,11	0,11	0,10	0,07	0,08	0,07	0,08	0,05	0,04	0,04	0,04	0,04
	Energy [%]	100	93,84	95,47	93,11	94,35	100	95,08	96,24	94,30	95,61	100	93,91	95,90	94,06	95,56

TABLE II: Experimental results for three spray velocities U and spray patterns (r, L) . In each run, the standard method ST is compared to 4 set-based approaches A-D in terms of time to complete spray task, length of end effector path, average end effector velocity and energy consumption.

		End effector avg. vel = 0.15 m/s					End effector avg. vel = 0.10 m/s					End effector avg. vel = 0.05 m/s				
		ST	A	B	C	D	ST	A	B	C	D	ST	A	B	C	D
r = 0.07 m L = 0.3 m	Time [s]	13,87	9,28	11,28	9,51	11,64	20,80	13,88	17,34	14,15	17,79	41,60	27,77	35,83	28,23	36,39
	Ee path [m]	2,08	1,39	1,67	1,40	1,73	2,08	1,39	1,72	1,40	1,77	2,08	1,39	1,78	1,42	1,81
	U [m/s]	0,15	0,22	0,18	0,22	0,18	0,10	0,15	0,12	0,15	0,12	0,05	0,08	0,06	0,07	0,06
	Energy [%]	100	66,37	82,07	67,18	83,17	100	65,02	82,99	66,41	84,47	100	65,62	85,62	66,67	87,03
r = 0.12 m L = 0.2 m	Time [s]	15,39	10,84	11,33	10,83	11,51	23,08	16,16	16,91	16,17	17,38	46,17	32,34	35,52	32,32	36,21
	Ee path [m]	2,31	1,62	1,70	1,62	1,72	2,31	1,62	1,69	1,61	1,73	2,31	1,62	1,74	1,62	1,78
	U [m/s]	0,15	0,21	0,20	0,21	0,20	0,10	0,14	0,14	0,14	0,13	0,05	0,07	0,07	0,07	0,06
	Energy [%]	100	67,67	73,71	67,63	73,22	100	65,84	70,98	65,48	71,90	100	66,70	75,52	66,92	76,16
r = 0.16 m L = 0.1 m	Time [s]	16,07	11,85	12,14	11,81	12,23	24,11	17,78	18,15	17,76	18,34	48,22	35,69	36,27	35,45	36,62
	Ee path [m]	2,42	1,76	1,83	1,77	1,84	2,42	1,78	1,82	1,78	1,83	2,42	1,77	1,81	1,77	1,83
	U [m/s]	0,15	0,20	0,20	0,20	0,20	0,10	0,14	0,13	0,14	0,13	0,05	0,07	0,07	0,07	0,07
	Energy [%]	100	73,68	73,69	70,28	72,96	100	71,28	73,32	70,54	73,45	100	71,92	72,73	70,11	73,05

TABLE III: Experimental results for three end effector velocities and spray patterns (r, L) . In each run, the standard method ST is compared to 4 set-based approaches A-D in terms of time to complete spray task, length of end effector path, spray trajectory velocity U and energy consumption.

of θ° . In approach A, mode switches are immediate, and as a result, there are peaks in the energy consumption and the joint velocities change abruptly. In approach C, mode 2 is activated at $\sigma_{FOV}(\mathbf{q}) = \theta - \theta_0 = 15^\circ$, and the transition between mode 1 and 2 is smoothed by the function in (50). This approach is more conservative, but there are no peaks in the energy consumption, and the joint velocities are much smoother. These two approaches are otherwise very similar, since the only difference is related to how mode switches are handled.

VI. CONCLUSION

This paper presents an algorithm to use set-based control in a spray paint scenario of a general surface. We exploit the fact that a small angle between the spray direction and the surface normal does not affect the quality of the paint, and define this angle as a set-based task with a maximum allowed value. An algorithm is developed to generate reference trajectories for a general robotic system that, if tracked, ensure achievement of the spray task and satisfaction of the set-based task. In other words, a desired spray pattern is tracked on the spray surface with some constant distance between the spray nozzle and the surface, and the maximum angle between the spray direction and the surface normal is never exceeded. In this paper, the spray pattern is represented as

a mowing-the-lawn pattern, but the proposed method is also applicable to other patterns.

Four different set-based approaches were implemented and experimentally verified for three different spray velocities and three spray patterns on a flat surface. Two of these approaches include smooth switching between different modes of the system. In doing so, it can no longer be guaranteed that the set-based task will always be satisfied, but the smooth switches ensure that the joint accelerations are not too large and are therefore suitable for larger spray velocities. Furthermore, in the experiments, the set-based task was never violated for any of the approaches.

Compared to the industry standard, the set-based approaches all consume less energy for the same spray trajectory velocity U . Furthermore, the set-based approaches require less torque in the turns, and is therefore applicable for higher spray velocities than the industry standard. By adjusting U so that the average end effector velocity is the same in all approaches, the set-based methods on average spent 74.50% of the time and 72.54% of the total energy compared to the current industry standard.

ACKNOWLEDGMENTS

This work was supported by the Research Council of Norway through the Center of Excellence funding scheme,

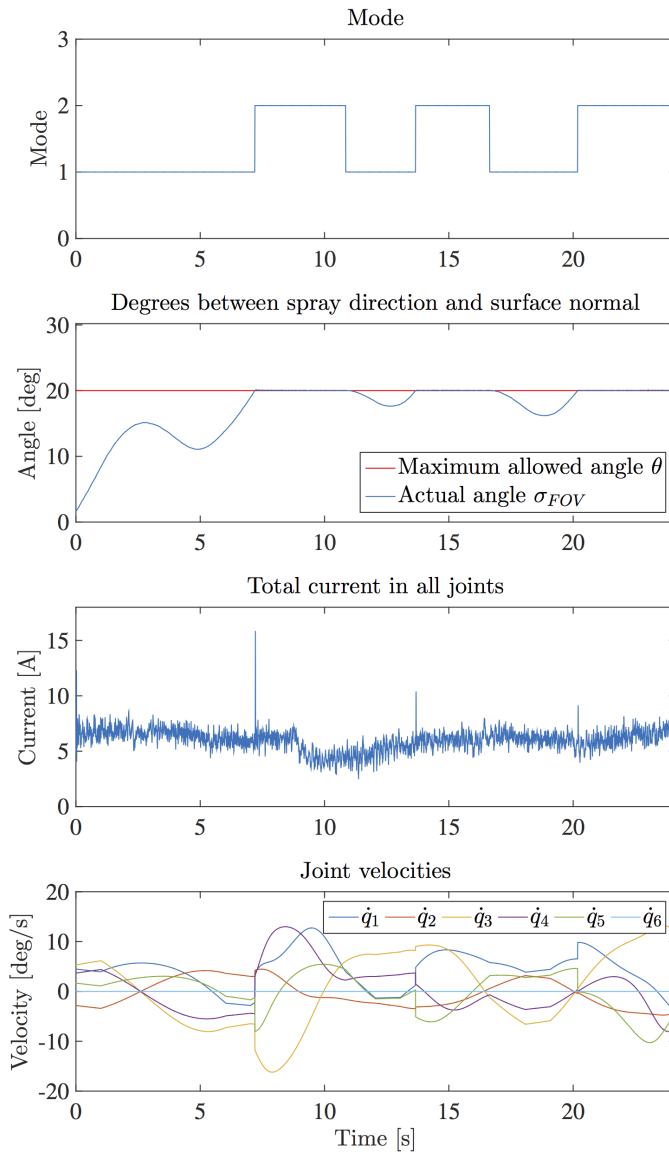


Fig. 12: Approach A, $U = 0.10$ m/s, $r = 0.16$ m, $L = 0.10$. Active mode over time, set-based task $\sigma_{FOV}(\mathbf{q})$, total current in all joints and joint velocities. In this approach, the set-based task σ_{FOV} evolves freely until the maximum limit θ , but the switch between modes results in spikes in current and large joint accelerations.

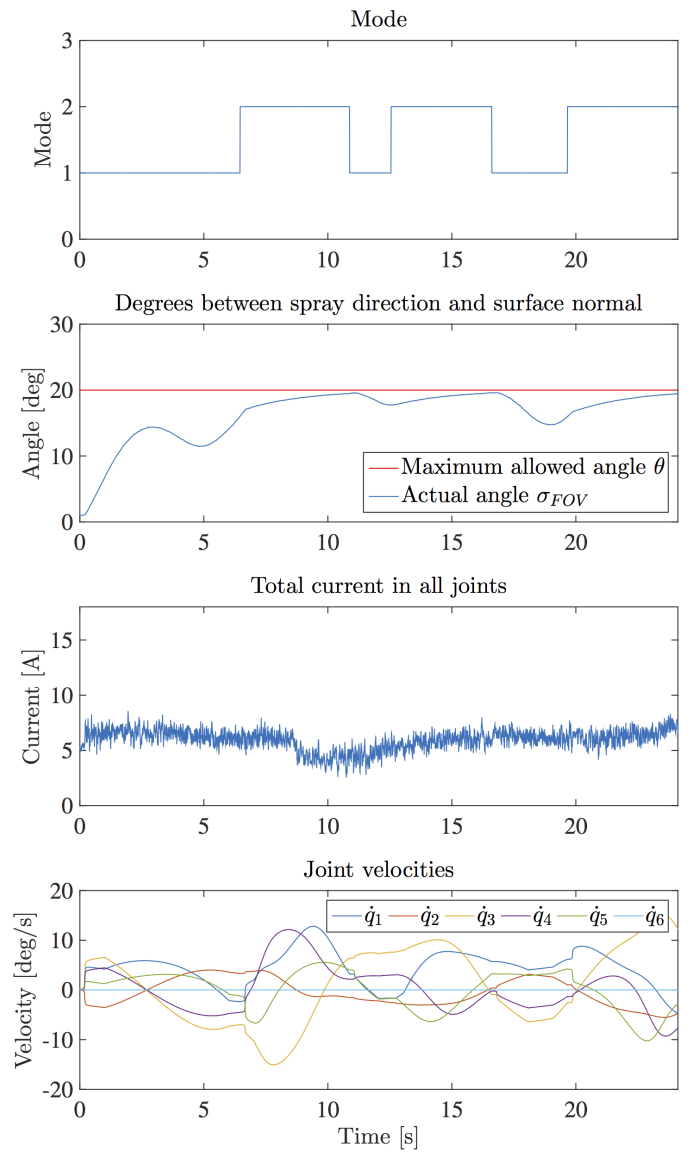


Fig. 13: Approach C, $U = 0.10$ m/s, $r = 0.16$ m, $L = 0.10$. Active mode over time, set-based task $\sigma_{FOV}(\mathbf{q})$, total current in all joints and joint velocities. In this approach, the set-based task σ_{FOV} evolves freely until a buffer limit ($\theta - \theta_0$), after which it is smoothly controlled towards the maximum limit. In this more conservative approach there are no large peaks in the current and the joint velocities are smooth.

project number 223254, and the centre for research based innovation SFI Manufacturing, project number 237900.

REFERENCES

- [1] F. Caccavale and B. Siciliano, "Kinematic control of redundant free-floating robotic systems," *Advanced Robotics*, vol. 15, no. 4, pp. 429–448, 2001.
- [2] O. Egeland and K. Y. Pettersen, "Free-floating robotic systems," in *Control Problems in Robotics and Automation*. Springer Berlin Heidelberg, 1998, vol. 230, pp. 119–134.
- [3] D. Nenchev, Y. Umetani, and Kazuya Y., "Analysis of a redundant free-flying spacecraft/manipulator system," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 1–6, 1992.
- [4] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," Tech. Rep., 2009. [Online]. Available: <https://www.math.ucsd.edu/sbuss/ResearchWeb/ikmethods/iksurvey.pdf>
- [5] C. A. Klein and C. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 2, pp. 245–250, 1983.
- [6] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [7] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Verlag, 2009.
- [8] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The null-space-based behavioral control for autonomous robotic systems," *Intelligent Service Robotics*, vol. 1, no. 1, pp. 27–39, 2008.

- [9] G. Antonelli, "Stability Analysis for Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 985–994, oct 2009.
- [10] G. Antonelli and S. Chiaverini, "Task-priority redundancy resolution for underwater vehicle-manipulator systems," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 1, no. May, 1998, pp. 768–773.
- [11] A. A. Maciejewski and C. A. Klein, "Obstacle Avoidance for Kinetically Redundant Manipulators in Dynamically Varying Environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, sep 1985.
- [12] D. Shiyou, Z. Xiaoping, and L. Guoqing, "The Null-Space-Based Behavioral Control for Swarm Unmanned Aerial Vehicles," in *Proc. 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control*. IEEE, oct 2011, pp. 1003–1006.
- [13] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," in *Proc. 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, vol. 1. IEEE, 2000, pp. 294–299.
- [14] Unicas, "Modeling and control for the MARIS UVMS, Tech. Rep. 5.
- [15] E. Marchand, F. Chaumette, and A. Rizzo, "Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 1996, pp. 1083–1090.
- [16] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Analysis and control of articulated robot arms with redundancy," in *Proc. 8th ZFAC World Congress*, 1981.
- [17] G. Antonelli, S. Moe, and K. Y. Pettersen, "Incorporating Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics," in *Proc. 23rd Mediterranean Conference on Control and Automation*, 2015.
- [18] S. Moe, A. R. Teel, G. Antonelli, and K. Y. Pettersen, "Stability Analysis for Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics Framework," in *Proc. 54th IEEE Conference on Decision and Control*, Osaka, Japan, 2015.
- [19] S. Moe, G. Antonelli, K. Y. Pettersen, and J. Schrimpf, "Experimental Results for Set-based Control within the Singularity-robust Multiple Task-priority Inverse Kinematics Framework," in *Proc. IEEE International Conference on Robotics and Biomimetics*, 2015.
- [20] P. J. From and J. T. Gravdahl, "A Real-Time Algorithm for Determining the Optimal Paint Gun Orientation in Spray Paint Applications," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 4, pp. 803–816, oct 2010.
- [21] S. H. Suh, I. K. Woo, and S. K. Noh, "Development of an automatic trajectory planning system (ATPS) for spray painting robots," in *Proc. 1991 IEEE International Conference on Robotics and Automation*. IEEE Comput. Soc. Press, 1991, pp. 1948–1955. [Online]. Available: <http://ieeexplore.ieee.org/document/131912/>
- [22] T. Kim and S. Sarma, "Optimal sweeping paths on a 2-manifold: a new class of optimization problems defined by path structures," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 613–636, aug 2003.
- [23] D. C. Conner, A. Greenfield, P. N. Atkar, A. A. Rizzi, and H. Choset, "Paint Deposition Modeling for Trajectory Planning on Automotive Surfaces," *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 4, pp. 381–392, oct 2005.
- [24] X. Li, O. A. Landsnes, H. Chen, S. M-V, T. A. Fuhlbrigge, and M. A. Rege, "Automatic Trajectory Generation for Robotic Painting Application," in *Proc. 41st International Symposium on Robotics and 6th German Conference on Robotics*, 2010, pp. 1–6.
- [25] Y. Tang and W. Chen, "Surface Modeling of Workpiece and Tool Trajectory Planning for Spray Painting Robot," *PLoS ONE*, vol. 10, no. 5, may 2015.
- [26] M. V. Andulkar, S. S. Chiddarwar, and A. S. Marathe, "Novel integrated offline trajectory generation approach for robot assisted spray painting operation," *Journal of Manufacturing Systems*, vol. 37, Part 1, pp. 201 – 216, 2015.
- [27] V. Potkonjak, G. S. Dordević, D. Kostić, and M. Rašić, "Dynamics of anthropomorphic painting robot: Quality analysis and cost reduction," *Robotics and Autonomous Systems*, vol. 32, no. 1, pp. 17–38, 2000.
- [28] H. Chen, W. Sheng, N. Xi, M. Song, and Y. Chen, "Automated robot trajectory planning for spray painting of free-form surfaces in automotive manufacturing," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 1, 2002, pp. 450–455.
- [29] H. Chen and N. Xi, "Automated Robot Tool Trajectory Connection for Spray Forming Process," *Journal of Manufacturing Science and Engineering*, vol. 134, 2012.
- [30] W. Chen and D. Zhao, "Path Planning for Spray Painting Robot of Workpiece Surfaces," *Mathematical Problems in Engineering*, vol. 2013, 2013.
- [31] P. J. From and J. T. Gravdahl, "General Solutions to functional and kinematic Redundancy," in *Proc. 2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 5779–5786. [Online]. Available: <http://ieeexplore.ieee.org/document/4434442/>
- [32] P. J. From, J. Gunnar, and J. T. Gravdahl, "Optimal Paint Gun Orientation in Spray Paint Applications - Experimental Results," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 438–442, apr 2011.
- [33] S. Moe, G. Antonelli, A. R. Teel, K. Y. Pettersen, and J. Schrimpf, "Set-Based Tasks within the Singularity-Robust Multiple Task-Priority Inverse Kinematics Framework: General Formulation, Stability Analysis, and Experimental Results," *Frontiers in Robotics and AI*, vol. 3, no. April, pp. 1–18, 2016. [Online]. Available: <http://journal.frontiersin.org/article/10.3389/frobt.2016.00016>
- [34] M. W. Spong and S. Hutchinson, *Robot Modeling and Control*. Wiley, 2005.
- [35] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins University Press, 1996.
- [36] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.
- [37] M. Asif, M. Junaid Khan, M. Safwan, and M. Rehan, "Feedforward and feedback kinematics controller for wheeled mobile robot trajectory tracking," *Journal of Automation and Control*, vol. 3, 2015.
- [38] H. Wu, W. Tizzano, T. T. Andersen, N. A. Andersen, and O. Ravn, "Hand-eye calibration and inverse kinematics of robot arm using neural network," *Advances in Intelligent Systems and Computing*, vol. 274, pp. 581–591, 2014.