rijksuniversiteit
groningen

Master Thesis

Dynamical compliant contact modeling of Van Gogh robotic manipulator

---

Author
O. (Orkhan) S. Allahverdiyev

Masters Research Project for Industrial
Engineering and Management

Faculty of science and engineering

Supervisor
prof. dr. ir. B. Jayawardhana

Second supervisor
M. (Mauricio) Muñoz Arias, PhD

April 22, 2022

Abstract

In recent years, artists and enthusiasts deployed robotic systems that not only replicate the already existing paintings but also develop novel forms of artistic paintings. Mimicking or even trying to match human creativity makes robot art paintings interesting and fascinating. However, fusing robotics with art naturally comes with challenges such as controlling the compliant contact of robot's end-effector before, during and after painting on a surface. In order to overcome these challenges, in this project we study a dynamical compliant contact model that incorporates the dynamical interaction of robot, brush, the transfer of the paint and compliant interaction between the brush at the end-effector and the canvas. The dynamical model is developed to enable the development of a compliant control and paint transfer control systems in order to achieve a smooth, uniform brushstroke on the canvas. We use Quanser 2-DOF serial link manipulator as a robotic system and model the paintbrush as a snubber mechanical system. The proposed model includes the viscous friction force during panting process, brush deformation and paint deposition rate from the paintbrush onto the canvas. We show that a simple PID-type controller can be added in order to achieve a desired painting speed and deformation speed of the brush. By applying a higher level control, the numerical simulations using Matlab/Simulink show that smooth, uniform brushstroke with a desired quality can be maintained.

Acknowledgement

I would like to thank my supervisor Bayu Jayawardhana. Without his supervision and understanding, the work that I have done would not have been possible.

List of Figures

List of Tables

Contents

1 Introduction

Due to the fast-paced development in robotics and automation technology, adoption of robots for artistic applications have been explored by artists and researchers. One of such applications is robotic painting. In recent years, artists and enthusiasts deployed robotic systems that not only replicate the already existing paintings but also develop novel forms of artistic paintings. Because it mimics or at the very least tries to match the human creativity, robotic painting creates lots of interest and fascination. One of the first such robotic system designed for this purpose was AARON, created in 70's as a plotter to produce artistic images [1]. Tresset and Leymarie created the robot Paul, which draws sketches of people using visual feedback [2]. e-David developed by Deuessen et al. is another example of such system, where based on an input image brushstrokes are created and then by help of a robotic arm these strokes are executed based on pen and ink [3]. Robotic system which is called Busker, produces watercolour paintings was created by Scalera et al [4]. It uses a collaborative robotic arm that can swap between brushes and watercolors when needed. Figure 1.1 illustrates the Busker robot in action and one of the paintings (Amanda) created by it. All these robotics painting systems mentioned above and many more unmentioned paved way to a competition called Robot Art, where various artists and enthusiasts compete against each other through their arts painted by the robotic systems [5].

(a) (b)

Fig. 1.1: a - Busker robot during painting process; b - the artwork, Amanda

Source: [6]

However, creating fully or even semi-autonomous robot painting system is not without challenges, firstly because it requires interdisciplinary knowledge (robotics and painting), human-robot interfacing, image processing and controlling the compliant contact of the robot's end effector with the canvas. Moreover, robot's movements and reach are subject to constraints due to it's end-effector task space. When designing such a system, following details and how to achieve control of them should be taken into account:

- the control of the robot's end effector position
- the angle at which the brush touches the canvas
- the control of the pressure of the brush onto the canvas
- the path of the brushstroke
- the painting speed
- controlling the quantity of paint deposited onto the canvas

To overcome these challenges, contact-based manipulation is required by the robot painting systems. Contact-based manipulation involves the interactive action of at least two dynamics, one originating from an object, structure, or other active system, and the other originating from the constraints - the dynamic environment. The dynamic interaction between the robot and the environment, which is often unpredictable, is a feature of contact tasks. The amount of mechanical work exchanged between the robot and the environment during contact can vary substantially in many circumstances, causing significant changes in the robotic control system's performance. As a result, either the interaction forces must be monitored and managed, or a control concept ensuring the robot's compliant interaction with the environment must be used to complete a contact task successfully [7].

1.1 Brushstroke Quality

Whether it is made by human or a robot, painting consists of collection of brushstrokes. Thus controlling the quality of each brushstroke is vital for the overall quality of the painting, especially for the robot painting system because controlling the quality of brushstrokes comes naturally to human. Human artist can intuitively assess the quality of the stroke and press brush harder against the canvas if the quality of the stroke is decreasing, squeezing more paint from the brush in process. Robots on the other hand, can only assess the quality of the brushstroke through image processing, so it is vital to get it right for the first time.

Figure 1.2 illustrates two different brushstrokes. Brushstroke on the left (a) is not uniform, meaning that paint quantity at the beginning of the stroke is greater than at the end of the stroke. Stroke on the right (b) however, can be claimed as an uniform, smooth brushstroke, meaning that paint deposition is consistent throughout the stroke. Question or rather a challenge arises, how do we achieve a uniform, smooth brushstrokes with robotic painting system? How we can overcome irregular, non-uniform (varying) brushstrokes in order to have a desired overall painting quality?

(a) (b)

Fig. 1.2: a - irregular brushstroke; b - smooth, uniform brushstroke

Source: [8]

1.2 Problem Statement

Based on the challenges in creating robotic painting systems stated earlier, we identify that there is a need to model the dynamical interaction between robot, paintbrush, the transfer of the paint and compliant interaction between the brush at the end-effector and the canvas. By doing so, we will be able to control the painting process and achieve the desired brushstroke characteristics like stroke length, width and quality. We can state the research problem that we are going to investigate as following:

How to achieve uniform, smooth brushstroke(s) during the painting process by a robotic painting system considering the (1) compliant interaction between the brush and the canvas, (2) dynamical interaction of robot and the brush, and (3) paint transfer from the brush onto the canvas.

1.3 Research Goal

Based on the problem statement and the research background that we already stated above, we can define the research goal as following:

"*Build a dynamical compliant contact model that enables the development of compliant control and paint transfer control in order to achieve painting smooth, uniform brushstroke on the canvas*"

1.4 Research Question(s)

Main research question is as following:

"*How we can dynamically model and control the robotic arm painting system in a way that it achieves painting smooth, uniform brushstrokes on a surface?*"

Three sub-questions are derived in order to answer the main research question:

- How to model the dynamic interaction of a robot manipulator and the painting brush?
- How to model the compliant contact between the painting brush and the canvas?
- How to control the painting and brush deformation speed in order to achieve the desired brushstroke quality?

1.5 Thesis Outline

Below is the outline of chapters and their contribution for the thesis.

Chapter 2: Related Work

Literature research is conducted about the previous robotic painting systems as well as the study of modeling the compliant contact between the robot manipulators and the environment.

Chapter 3: Dynamical Model

The main scope of this thesis is to build a dynamical model consisting of robot dynamics, brush dynamics and paint transfer system and our proposed dynamical model will be presented in this chapter. Dynamical model will include robot's constraint dynamics, brush dynamics, friction and paint deposition model.

Chapter 4: Simulation Set-up

We will discuss about how we setup the simulation model in order to verify the proposed dynamical model. Design parameters for simulation runs will be discussed in here.

Chapter 5: Results

In this chapter, we will present the results of those simulation runs along with the proposed PID-controllers for painting speed, deformation speed and a higher-level control for painting quality.

Chapter 6: Conclusion and Discussion

Results of the simulations will be analyzed in this chapter. We will also present the limitations of our proposed model and make a list of suggestions to improve the current work.

Related Work

2

2.1 Robot-Environment Contact

This section contains background information on contact forces resulting from robot-environment interaction. In the literature, modeling the robot-environment interaction is divided into two main categories, interaction with rigid environment and with flexible environments. Rigid-body modeling assumes that no deformations are allowed at the surfaces of contact between bodies. Instead, contact forces arise from two sources: the constraint due to the impenetrability of the surface and surface frictional forces. To give an example for this, we can show the modeling of writing task by Veljko et al., where the robot environment is considered in the form of geometric constraints [9].

Unlike rigid-body model, a compliant contact deforms under the influence of applied forces.The forces of interaction at the contact are derived from the compliance or stiffness model.

2.1.1 Compliant Contact

In their research, Verscheure et al. analyze the contact dynamics between the robot manipulator and the environment [10]. They model the compliance of the environment in the normal direction, meaning that tangential forces to the environment are due to the friction. The force-deformation relation is described by a simple linear spring model

$$F_n(\delta) = \begin{cases} \kappa\delta, & \text{if } \delta > 0, \\ 0, & \text{if } \delta \leq 0, \end{cases}$$

where δ and κ are the environment deformation and stiffness. As for the friction model, a Coulomb model is used

$$F_f\left(v_t\right) = \begin{cases} \text{undefined}, & \text{if } \left|v_t\right| < v_{t,m} \\ -\operatorname{sgn}\left(v_t\right)\left(\mu F_n\right), & \text{if } \left|v_t\right| \geq v_{t,m}, \end{cases}$$

where v_t is the relative tangential velocity in the contact, $\mathrm{sgn}(\cdot)$ is defined as the signum function, μ is the coefficient of friction, $v_{t,m}$ is a threshold velocity.

Baptista et al. on the other hand, argue that robot-environmental contact behaves in a non-linear way, thus a non-linear spring/damper model is better suited to model it [11]. Figure 2.1 depicts a robot-manipulator constrained by contact with the environment. Robot dynamics in Cartesian space proposed by authors is as following:

$$M_x(x)\ddot{x} + C_x(x, \dot{x})\dot{x} + g_x(x) + d_x(\dot{x}) = f - f_e$$

where x denotes the vector of the position and orientation of the manipulator's end-effector, f is the robot's input force, f_e is the contact force vector and J represents the Jacobian matrix.

$f_e = \begin{bmatrix} f_n & f_t \end{bmatrix}^T$ denotes the interaction force vector and is composed of tow forces, normal contact force f_n and the tangential contact forces f_t caused by friction contact between the end-effector and the environment. The normal contact force f_n is modeled as a nonlinear spring-damper mechanical system according to [12]:

$$f_n = ke\delta x + \rho_e(\delta x)\dot{x}$$

where the terms k_e and ρ_e are the environment stiffness and damping coefficients, respectively, $\delta x = x - x_e$ is the penetration depth, and x_e stands for the distance between the surface and the base Cartesian frame. The tangential contact force vector f_t due to surface friction is assumed to be given as proposed by [13]:

$$f_t = \mu \left| f_n \right| \mathrm{sgn}\left(\dot{x}_p \right)$$

where $\dot{x}_p$ is the sliding velocity and μ is the dry friction coefficient between the end-effector and the contact surface.

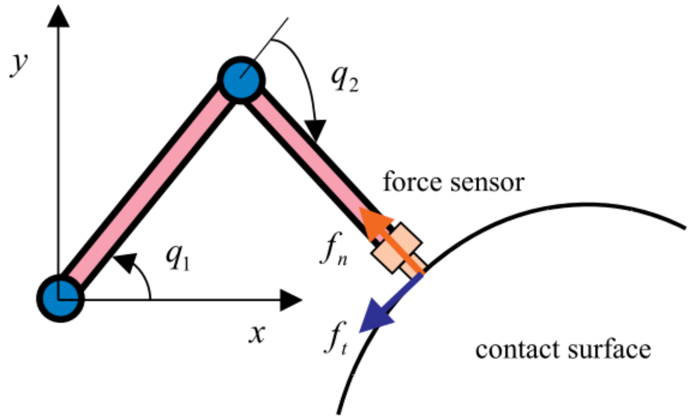

Fig. 2.1: Robot Manipulator applying a desired force on the environment

Source: [11]

2.2 Brush Modeling

In this section, we take a look at various paintbrush modeling examples in the literature. Modeling the paintbrush and it's interaction with the canvas is essential in designing the robotic painting system.

Otsuki et al. designed and developed a brush device with mechanisms that provide the sensation of painting [14]. It is a mixed reality painting system meaning that user holds the physical device in real world and paints in virtual environment. According to authors, the painting operation can be categorized into two following actions: pushing the brush tip onto the canvas and stroking the brush tip across the canvas. During the pushing the brush tip onto the canvas action, user perceives the pushing sensation by the amount of brush tip bending and the reaction forces from the canvas. During the stroking the brush on the canvas action, frictional forces are generated in the opposite direction to the stroking movement. User perceives this sensation by the direction of tip bending and the direction of the force. Frictional force generated during stroking action depend on the canvas material, and the brush tip dryness, which is the amount of paint left on the brush tip.

Considering these forces, authors develop a paintbrush model that includes the bending of the brush tip by amount and the direction, the reaction force due to the canvas, and the frictional forces between the brush tip and the canvas. Figure 2.2 below depicts the illustration of brush tip bending during the painting process and the resulting forces. Brush tip is modeled as spring and when brush tip bends, an elastic force tries to recover the brush tip to it's original shape. Elastic force is dependent on the angle of brush tip bending (θ) and brush tip hardness (k) and formulated as following, $F_e = k\theta$. Frictional force, on the other hand, formulated as $F_f = \mu_s N$ and where μ_s is static friction coefficient and N is the normal force. Authors develop model further by integrating weighted-function, $g(v)$, for brush tip dryness in order to reflect the change of friction force depending on the amount of paint in the brush tip.

$$g(v) = 1 - (v/V_{\max})$$

where v is the current amount of water and $V_{\max}$ is the maximum amount of water. This reflects in the calculation of friction coefficient, μ_1, where α is the minimal friction coefficient that exists in every material and μ_s is the dry friction coefficient.

$$\mu_1 = (\mu_s - \alpha)g(v) + \alpha$$

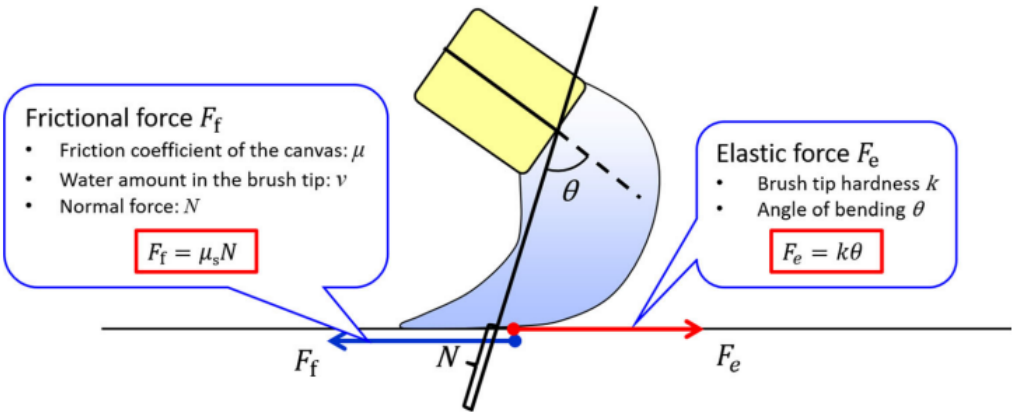

Fig. 2.2: Elastic and frictional force when brush tip is bending

Source: [14]

Baxter et al. developed a deformable, physically-based 3D brush model with a haptic feedback, which gives the user control of complex brush strokes [15]. Brush head was modeled as a subdivision surface mesh wrapped around a spring-mass particle system skeleton. Brush force is modeled as a linear function of the penetration depth of the undeformed brush point. If we take

d_p as the penetration depth , and l_p as the length of the brush head projected onto the canvas normal, $\mathbf{n}$, then the force is modeled as:

$$\mathbf{f}_b\left(d_p\right) = \begin{cases} 0 & \text{if } d_p \leq 0 \\ \mathbf{n}\left(k_1/l_p\right) d_p & \text{if } 0 < d_p \leq l_p \\ \mathbf{n}\left(k_1 + \left(k_2/l_p\right)\left(d_p - l_p\right)\right) & \text{if } l_p < d_p \end{cases}$$

where k_1 is a constant modeling spring of bristles and k_2 is a larger positive constant that simulates collision of the actual brush handle with the canvas. The value of k_1 determines the brush stiffness. Friction force, $\mathbf{f}_t$, is modeled as a force opposite the current brush velocity, $\mathbf{v}_b$:

$$\mathbf{f}_t = k_t(\mathbf{v}_b - \mathbf{n}(\mathbf{n}\mathbf{v}_b))$$

where k_t is the coefficient of the friction.

Guo et al. propose a novel simulation of the brush stroke by apply force feedback technology to the virtual painting process [16]. Brush is constructed as spring-mass model and the relationship between force and the brush deformation is analyzed. As we can see in the Figure 2.3 below, brush spring is perpendicular to the paper plane and when the pressure is exerted on the brush, the spring moves downward. The feedback pressure (F) is proportional to the downward displacement of the brush and formulate as following:

$$F = \lambda H X$$

where λ is the force feedback factor which controls the magnitude of F and value of is related to hardware and determined through experiments. The unit of λ is $\mathrm{N/mm}$. Brush hardness factor is denoted by H and take values $H \in (0,1)$. The larger the value of H is, the harder the brush is; thus, the exerted force is larger when the brush moves down the unit displacement. X is the downward displacement of the brush and also represents the deformation amount of the spring and it's is mm. The friction force (F_f) between the brush and paper is proportional to the pressure, and mathematically expressed by authors as following:

$$F_f = \mu \cdot F,$$

where F is the feedback pressure, and μ is friction coefficient.

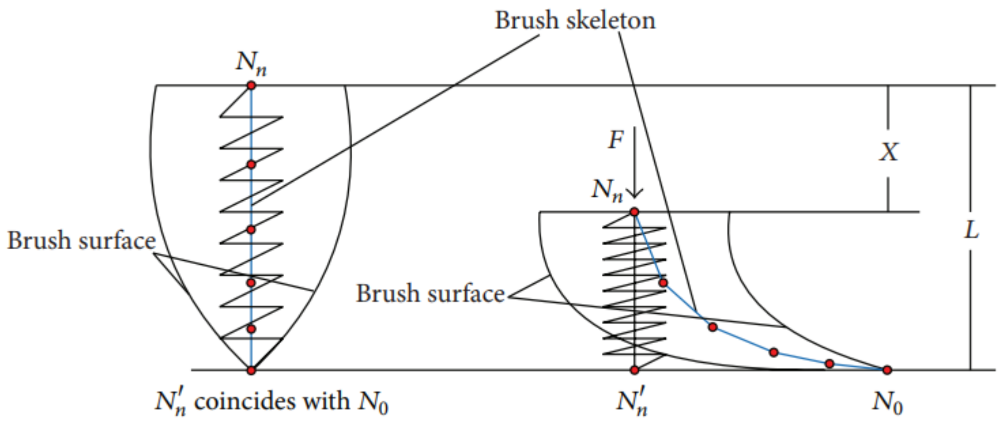

Fig. 2.3: The spring mass model of the brush tip

Source: [16]

In a simulation experiment, reasonable hardness factors determined to be about 0.3-0.7. It was estimated that μ value range is about 0.2-0.03. By using these values, effects of the brush strokes under the pressure of different magnitude and painting techniques were experimented. We can see the results of it in Figure 2.4. As we can see, range for feedback force was between 2-3.5N.

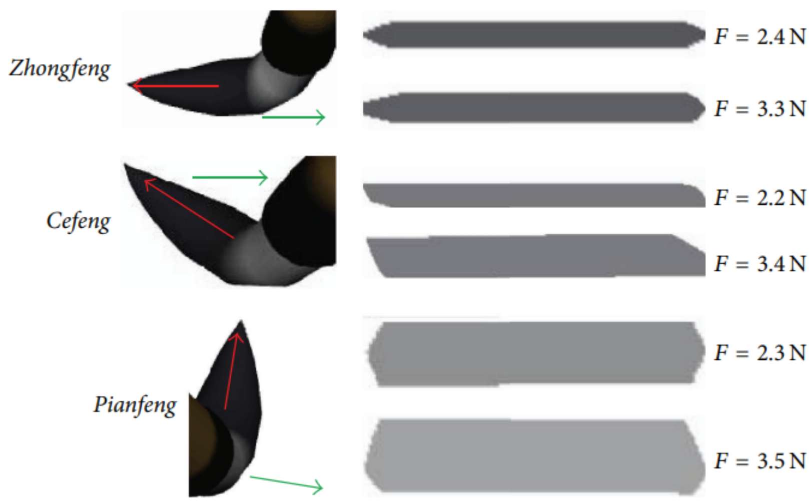

Fig. 2.4: Effects of varying pressure on the brush strokes

Source: [16]

Scalera et al. study the influence of the z-coordinate of the brush and painting speed on the brush strokes (Figure 2.5). Experiments were conducted on a robotic painting system called Busker, which is a is a 6-DOF UR10 robot [17].

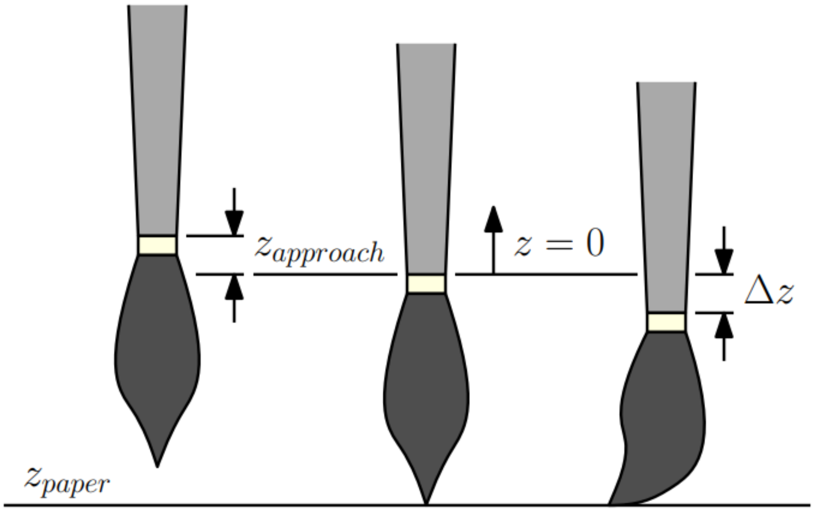

Fig. 2.5: Brush positioning on the painting paper

Source: [17]

During experiments, linear strokes (250mm long) has been painted by the robot while varying the z-coordinate from $z_1 = 0$ mm to $z_3 = -2$ mm with $\Delta z = 1$ mm, and the maximum robot speed, from $v_1 = 0.1$ m/s to $v_3 = 0.3$ m/s with $\Delta v = 0.1$ m/s. It was shown that high painting speed can cause undesirable effects in the starting and end points of the stroke, due to rough landing and rough separation. As expected, stroke thickness the z-coordinate of the brush affects the stroke thickness, since more bristles are pressed against the paper, the larger the stroke thickness. The stroke thickness shows a decreasing trend along the stroke.

2.3 Research Contribution

While both robot-environment contact and paintbrush modeling are available in the literature, there is a gap regarding the dynamics of both modeling at the same time. In our research, we will try to bridge this gap by proposing a dynamical model where we incorporate the compliant modeling between the brush and canvas as well as robot and brush dynamics. Paint transfer from brush onto the canvas and it's effect on the surface friction force will also be considered in the proposed dynamical model.

3 Proposed Dynamical Model

In this chapter we introduce a painting robotic system, which we call Van Gogh painting robot and a dynamical model for that robot. Our robotic system consists of 2-DOF RR robotic manipulator and a painting brush attached to it's end-effector. Figure 3.1 illustrates the Van Gogh painting robot. We can divide the painting process of van Gogh robot into three phases. At first phase, robot's end-effector approaches the canvas. Then tip of the painting brush makes contact with the canvas. And finally, brush slides along the canvas, making brushstrokes in the process.

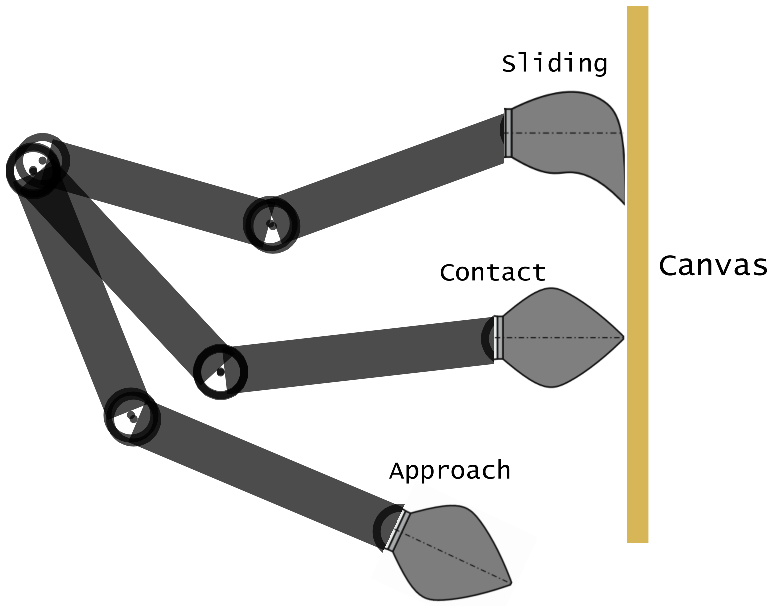

Fig. 3.1: Painting robot, Van Gogh approaching, making contact and stroking on the canvas

In order to propose a complete dynamical compliant model for the Van Gogh painting robot, we should analyze carefully the three phases and what dynamics play role in each of these phases. During the approach phase, robot's own dynamics play a role. It drives the painting brush towards the canvas. When brush tip makes initial contact with the canvas, brush dynamics and robot's constrained dynamics to not let the brush punch through canvas is activated. Paint transfer model and friction force between the brush and the canvas plays a role in the sliding-stroking phase.

Thus, we will start the chapter with the modeling of robot dynamics and move into constrained dynamics. We will first introduce the general robot dynamics and then apply it to our proposed 2RR robotic manipulator. Afterwards we will introduce the brush modeling and conclude the chapter with a proposed paint transfer and friction force model.

3.1 Robot Dynamics

Before we move into robot dynamics, we will first introduce some important terms that will be useful to understand robot's dynamics and proposed constraint dynamics.

3.1.1 Forward Kinematics & Jacobian

Calculation of the position and orientation of robot's end-effector frame from its joint coordinates is done by forward kinematics. Figure 3.2 illustrates the forward kinematics problem of a 2RR planar open chain. The robot arm lengths are given by L_1, L_2. The task-space, Cartesian, position (x, y) and orientation of the end-effector frame as functions of joint angles are given by the following equations

$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \tag{3.1}$$

$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \tag{3.2}$$

$$\phi = \theta_1 + \theta_2 \tag{3.3}$$

Equation (3.3) represents the orientation of the end-effector and can be omitted if we are interested in the (x, y) position of the end-effector.

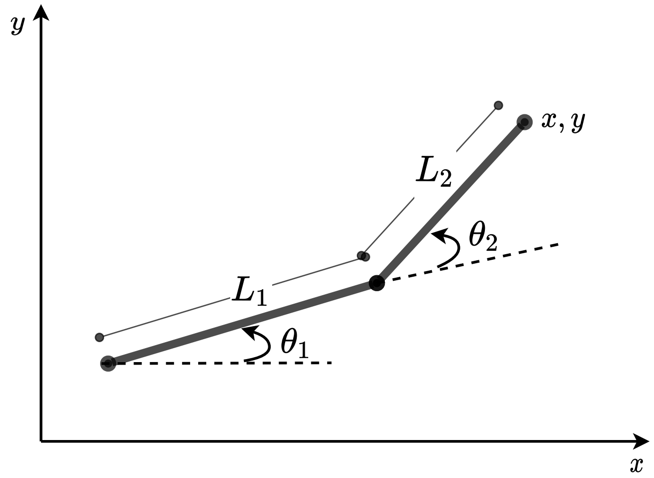

Fig. 3.2: Forward Kinematics of 2RR manipulator

The velocity of end-effector is denoted by the equation $\dot{x} = dx/dt$. In this case, we can write the forward kinematics as following

$$x(t) = f(\theta(t)) \tag{3.4}$$

Where $\theta \in \mathbf{R}^n$ is a set of joint variables. By the chain rule, the time derivative at time t is

$$\dot{x} = \frac{\partial f(\theta)}{\partial \theta}\frac{\partial \theta(t)}{\partial t} = \frac{\partial f(\theta)}{\partial \theta}\dot{\theta} = J(\theta)\dot{\theta} \tag{3.5}$$

where $J(\theta) \in \mathbf{R}^{m\times n}$ is called the Jacobian. For 2RR manipulator, differentiating the Equations (3.1) and (3.2) with respect to time yields

$$\dot{x} = -L_1\dot{\theta}_1 \sin\theta_1 - L_2(\dot{\theta}_1 + \dot{\theta}_2)\sin(\theta_1 + \theta_2)$$

$$\dot{y} = L_1\dot{\theta}_1 \cos\theta_1 + L_2(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1 + \theta_2)$$

Which in turn can be rearranged into an equation form $\dot{x} = J(\theta)\dot{\theta}$:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}-L_1\sin\theta_1 - L_2\sin(\theta_1+\theta_2) & -L_2\sin(\theta_1+\theta_2)\\ L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2) & L_2\cos(\theta_1+\theta_2)\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} \tag{3.6}$$

Writing the two columns of $J(\theta)$ as $J_1(\theta)$ and $J_2(\theta)$ and the tip velocity $\dot{x}$ as v_{tip}, Equation (3.6) becomes

$$v_{\text{tip}} = J_1(\theta)\dot{\theta}_1 + J_2(\theta)\dot{\theta}_2 \tag{3.7}$$

We can observe that $J_1(\theta)$ and $J_2(\theta)$ depend on the joint values θ_1 and θ_2 thus one may ask whether there are any set of configurations at which $J_1(\theta)$ and $J_1(\theta)$ become collinear. To give one such example: when θ_2 is 0°or 180°then, regardless of the value of θ_1, $J_1(\theta)$ and $J_2(\theta)$ will be collinear and $Jacobian J(\theta)$ becomes a singular matrix. Therefore such configurations are called **singularities**; when singularities occur the robot tip is cannot generate velocities in certain directions.

The Jacobian matrix is essential when modeling external application to the robot's end-effector. If we take the tip force vector generated by the robot as f_{tip} and the joint torque vector by τ , the conservation of power then requires that

$$f_{\text{tip}}^{\text{T}} v_{\text{tip}} = \tau^{\text{T}} \dot{\theta} \tag{3.8}$$

for all arbitrary joint velocities $\dot{\theta}$. Since $v_{\text{tip}} = J(\theta)\dot{\theta}$, the equality

$$f_{\text{tip}}^{\text{T}} J(\theta)\dot{\theta} = \tau^{\text{T}} \dot{\theta} \tag{3.9}$$

must hold for all possible $\dot{\theta}$. This can only be true if

$$\tau = J^{\text{T}}(\theta) f_{\text{tip}} \tag{3.10}$$

The joint torque τ needed to create the tip force f_{tip} is calculated from the equation above.

Applying this to our 2-DOF planar manipulator, $J(\theta)$ is a square matrix dependent on θ. If there is no singularity case then both $J(\theta)$ and $J^{\text{T}}(\theta)$ are invertible, thus Equation (3.10) can be written in the following form:

$$f_{\text{tip}} = ((J(\theta))^{\text{T}})^{-1}\tau = J^{-\text{T}}(\theta)\tau \tag{3.11}$$

3.1.2 Robot Dynamics

The subject of robot dynamics is to study the motions of robots, taking into account the forces and torques that cause these motions. The associated dynamic equations – also referred to as the equations of motion – are a set of second-order differential equations of the form

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) \tag{3.12}$$

where $\theta \in \mathbb{R}^n$ is the vector of joint variables, $\tau \in \mathbb{R}^n$ is the vector of joint forces and torques, $M(\theta) \in \mathbb{R}^{n \times n}$ is a symmetric positive-definite mass matrix, and $h(\theta, \dot{\theta}) \in \mathbb{R}^n$ are forces that combine together centripetal, Coriolis, gravity, and friction terms that depend on θ and $\dot{\theta}$.

Using this equation, we can derive a formula to determine the robot's acceleration $\ddot{\theta}$ given the state $(\theta, \dot{\theta})$ and the joint forces and torques,

$$\ddot{\theta} = M^{-1}(\theta)(\tau - h(\theta, \dot{\theta})) \tag{3.13}$$

We can refer to Equation (3.13) as forward dynamics and Equation (3.12) as inverse dynamics.

We can use Lagrangian formulation of dynamics to derive the robot's dynamic equation. A Lagrangian function $\mathcal{L}(q, \dot{q})$ is defined in [18] as the overall system's kinetic energy $\mathcal{K}(q, \dot{q})$ minus the potential energy $\mathcal{P}(q)$

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{P}(q)$$

The equations of motion can now be expressed in terms of the Lagrangian as follows:

$$f = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q},$$

The equation of motion is then given by

$$f = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} - \frac{\partial \mathcal{L}}{\partial x} = \mathfrak{m}\ddot{x} + \mathfrak{m}g$$

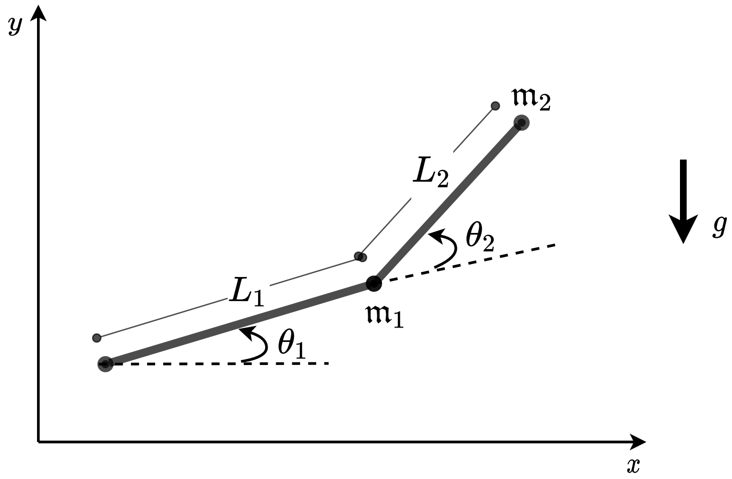

Fig. 3.3: 2RR planar manipulator under gravity

If we take a look at the $2\mathrm{RR}$ manipulator with the presence of gravity (Figure 3.3), we can see that the robot moves in the $\hat{x}-\hat{y}$-plane, with gravity g acting in the $-\hat{y}$-direction. The position and velocity of the link-1 mass are then given by

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} L_1 \cos\theta_1 \\ L_1 \sin\theta_1 \end{bmatrix}$$
$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix} = \begin{bmatrix} -L_1 \sin\theta_1 \\ L_1 \cos\theta_1 \end{bmatrix} \dot{\theta}_1,$$

while those of the link-2 mass are given by

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} L_1 \cos\theta_1 + L_2 \cos(\theta_1+\theta_2) \\ L_1 \sin\theta_1 + L_2 \sin(\theta_1+\theta_2) \end{bmatrix},$$
$$\begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} -L_1 \sin\theta_1 - L_2 \sin(\theta_1+\theta_2) & -L_2 \sin(\theta_1+\theta_2) \\ L_1 \cos\theta_1 + L_2 \cos(\theta_1+\theta_2) & L_2 \cos(\theta_1+\theta_2) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}.$$

The Lagrangian $\mathcal{L}(\theta,\dot{\theta})$ can be expressed as of the form

$$\mathcal{L}(\theta,\dot{\theta}) = \sum_{i=1}^{2} (\mathcal{K}_i - \mathcal{P}_i)$$

where the link kinetic energy terms $\mathcal{K}_1$ and $\mathcal{K}_2$ are

$$
\begin{aligned}
\mathcal{K}_1 &= \frac{1}{2}\mathfrak{m}_1\left(\dot{x}_1^2+\dot{y}_1^2\right)=\frac{1}{2}\mathfrak{m}_1 L_1^2\dot{\theta}_1^2\\
\mathcal{K}_2 &= \frac{1}{2}\mathfrak{m}_2\left(\dot{x}_2^2+\dot{y}_2^2\right)\\
&= \frac{1}{2}\mathfrak{m}_2\left(\left(L_1^2+2L_1L_2\cos\theta_2+L_2^2\right)\dot{\theta}_1^2+2\left(L_2^2+L_1L_2\cos\theta_2\right)\dot{\theta}_1\dot{\theta}_2+L_2^2\dot{\theta}_2^2\right)
\end{aligned}
$$

and the link potential energy terms $\mathcal{P}_1$ and $\mathcal{P}_2$ are

$$
\begin{aligned}
\mathcal{P}_1 &= \mathfrak{m}_1 g y_1 = \mathfrak{m}_1 g L_1 \sin\theta_1,\\
\mathcal{P}_2 &= \mathfrak{m}_2 g y_2 = \mathfrak{m}_2 g\left(L_1\sin\theta_1+L_2\sin\left(\theta_1+\theta_2\right)\right)
\end{aligned}
$$

Explicit dynamic equations for the $2\mathrm{RR}$ manipulator is as following:

$$
\begin{aligned}
\tau_1 =& \left(\mathfrak{m}_1 L_1^2+\mathfrak{m}_2\left(L_1^2+2L_1L_2\cos\theta_2+L_2^2\right)\right)\ddot{\theta}_1\\
&+\mathfrak{m}_2\left(L_1L_2\cos\theta_2+L_2^2\right)\ddot{\theta}_2-\mathfrak{m}_2L_1L_2\sin\theta_2\left(2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_2^2\right)\\
&+\left(\mathfrak{m}_1+\mathfrak{m}_2\right)L_1 g\cos\theta_1+\mathfrak{m}_2 g L_2\cos\left(\theta_1+\theta_2\right),\\
\tau_2 =& \mathfrak{m}_2\left(L_1L_2\cos\theta_2+L_2^2\right)\ddot{\theta}_1+\mathfrak{m}_2L_2^2\ddot{\theta}_2+\mathfrak{m}_2L_1L_2\dot{\theta}_1^2\sin\theta_2\\
&+\mathfrak{m}_2 g L_2\cos\left(\theta_1+\theta_2\right).
\end{aligned}
$$

Gather terms together results in an equation of the form

$$
\tau = M(\theta)\ddot{\theta}+\underbrace{c(\theta,\dot{\theta})+g(\theta)}_{h(\theta,\dot{\theta})},
$$

with

$$
M(\theta)=\begin{bmatrix}\mathfrak{m}_1L_1^2+\mathfrak{m}_2\left(L_1^2+2L_1L_2\cos\theta_2+L_2^2\right) & \mathfrak{m}_2\left(L_1L_2\cos\theta_2+L_2^2\right)\\ \mathfrak{m}_2\left(L_1L_2\cos\theta_2+L_2^2\right) & \mathfrak{m}_2L_2^2\end{bmatrix},
$$

$$
c(\theta,\dot{\theta})=\begin{bmatrix}-\mathfrak{m}_2L_1L_2\sin\theta_2\left(2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_2^2\right)\\ \mathfrak{m}_2L_1L_2\dot{\theta}_1^2\sin\theta_2\end{bmatrix},
$$

$$
g(\theta)=\begin{bmatrix}\left(\mathfrak{m}_1+\mathfrak{m}_2\right)L_1g\cos\theta_1+\mathfrak{m}_2gL_2\cos\left(\theta_1+\theta_2\right)\\ \mathfrak{m}_2gL_2\cos\left(\theta_1+\theta_2\right)\end{bmatrix},
$$

where $M(\theta)$ is the symmetric positive-definite mass matrix, $c(\theta,\dot{\theta})$ is the vector containing the Coriolis and centripetal torques, and $g(\theta)$ is the vector containing the gravitational torques.

These dynamic equations represent the situation when we chose the links at the ends of each link as point masses $\mathfrak{m}_1$ and $\mathfrak{m}_2$ concentrated at the ends of each link. For comparison, if we chose the masses of each link to be at the center of corresponding link, our dynamical equations, derived from [19], will be as:

$$M(\theta) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

with

$$\begin{aligned} M &= I_{o1} + \mathfrak{m}_1 L_1^2/4 + I_{o2}\mathfrak{m}_2 L_2^2/4 + \mathfrak{m}_2 L_1^2 + \mathfrak{m}_2 L_1 L_2 \cos\theta_2, \\ M_{12} &= M_{21} = I_{o2} + \mathfrak{m}_2 L_2^2/4 + \mathfrak{m}_2 L_1 L_2 \cos\theta_2/2, \\ M_{22} &= I_{o2} + \mathfrak{m}_2 L_2^2/4, \end{aligned}$$

where I_{oi} denotes the inertia of link i in kgm^2.

Vector containing the Coriolis and centripetal torques is given by the equation

$$c(\theta,\dot{\theta}) = \mathcal{D}(\theta,\dot{\theta})\dot{\theta}$$

where Coriolis torque matrix $\mathcal{D}(q,\dot{q})$ is as following

$$\mathcal{D}(\theta,\dot{\theta}) = \begin{bmatrix} \mathcal{D}_{11} & \mathcal{D}_{12} \\ \mathcal{D}_{21} & \mathcal{D}_{22} \end{bmatrix}$$

with

$$\begin{aligned} \mathcal{D}_{11} &= -\left(\mathfrak{m}_2 L_1 L_2 \sin\theta_2/2\right)\dot{\theta}_2 \\ \mathcal{D}_{12} &= -\left(\mathfrak{m}_2 L_1 L_2 \sin\theta_2/2\right)\left(\dot{\theta}_1 + \dot{\theta}_2\right) \\ \mathcal{D}_{21} &= \left(\mathfrak{m}_2 L_1 L_2 \sin\theta_2/2\right)\dot{\theta}_1 \\ \mathcal{D}_{22} &= 0 \end{aligned}$$

The gravity vector $g(q)$ is given by

$$g(\theta) = \begin{bmatrix} \mathfrak{m}_1 g_0 L_1 \cos\theta_1/2 + \mathfrak{m}_2 g_0 L_1 \cos\theta_1 + \mathfrak{m}_2 g_0 L_2 \cos\left(\theta_1+\theta_2\right)/2 \\ \mathfrak{m}_2 g_0 L_2 \cos\left(\theta_1+\theta_2\right) \end{bmatrix}$$

3.1.3 Constrained Dynamics

If we consider the case of a painting robot, we can think of the canvas as a geometric constraint. Because by design, robot's end-effector cannot punch through the canvas, so it should make contact and then slide across the canvas during the painting process. We can model it as a set of k holonomic or nonholonomic Pfaffian velocity constraints of the form

$$A(\theta)\dot{\theta} = 0, \quad A(\theta) \in \mathbb{R}^{k \times n} \tag{3.14}$$

Our assumption is that the constraints do no work on the robot, i.e., the generalized forces τ_{con} due to the constraints satisfy

$$\tau_{\text{con}}^{\mathrm{T}}\dot{\theta} = 0.$$

This assumption means that τ_{con} must be a linear combination of the columns of $A^{\mathrm{T}}(\theta)$, i.e., $\tau_{\text{con}} = A^{\mathrm{T}}(\theta)\lambda$ for some $\lambda \in \mathbb{R}^k$, since these are the generalized forces that do no work when $\dot{\theta}$ is subject to the constraints as derived in [18]:

$$\left(A^{\mathrm{T}}(\theta)\lambda\right)^{\mathrm{T}}\dot{\theta} = \lambda^{\mathrm{T}}A(\theta)\dot{\theta} = 0 \quad \text{for all } \lambda \in \mathbb{R}^k.$$

Adding the constraint forces $A^{\mathrm{T}}(\theta)\lambda$ to the robot dynamics, we can formulate the constrained equations of motion as

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) + A^{\mathrm{T}}(\theta)\lambda \tag{3.15}$$

$$A(\theta)\dot{\theta} = 0 \tag{3.16}$$

where λ is a set of Lagrange multipliers and $A^{\mathrm{T}}(\theta)\lambda$ are the forces applied against the constraints as expressed as joint forces and torques.

Lagrange multipliers were derived in [18] as following:

$$\lambda = \left(AM^{-1}A^{\mathrm{T}}\right)^{-1}\left(AM^{-1}(\tau - h) + \dot{A}\dot{\theta}\right) \tag{3.17}$$

Using these multipliers we try to derive the end-effector tip force in task space.

$$J^{\mathrm{T}}(\theta)\mathcal{F}_{\text{tip}} = A^{\mathrm{T}}(\theta)\lambda \tag{3.18}$$

If $J(\theta)$ is invertible, meaning that there is no singularity then

$$\mathcal{F}_{\text{tip}} = J^{-\mathrm{T}}(\theta)A^{\mathrm{T}}(\theta)\lambda \tag{3.19}$$

When we apply the constrained dynamics equation that are derived above to 2RR manipulator and set the tip of the robot at (x, y), then the robot's forward kinematics can be written as following

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1 \mathrm{c}_1 + L_2 \mathrm{c}_{12} \\ L_1 \ \mathrm{s}_1 + L_2 \mathrm{s}_{12} \end{bmatrix},$$

where s_{12} and c_{12} are $\sin(\theta_1 + \theta_2)$ and $\cos(\theta_1 + \theta_2)$, respectively. The derivatives of the forward kinematics are

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \underbrace{\begin{bmatrix} -L_1 \mathrm{s}_1 - L_2 \mathrm{s}_{12} & -L_2 \mathrm{s}_{12} \\ L_1 \mathrm{c}_1 + L_2 \mathrm{c}_{12} & L_2 \mathrm{c}_{12} \end{bmatrix}}_{J(\theta)} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix},$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = J(\theta)\ddot{\theta} + \underbrace{\begin{bmatrix} -L_1 \dot{\theta}_1 \mathrm{c}_1 - L_2 \left(\dot{\theta}_1 + \dot{\theta}_2\right) \mathrm{c}_{12} & -L_2 \left(\dot{\theta}_1 + \dot{\theta}_2\right) \mathrm{c}_{12} \\ -L_1 \dot{\theta}_1 \ \mathrm{s}_1 - L_2 \left(\dot{\theta}_1 + \dot{\theta}_2\right) \mathrm{s}_{12} & -L_2 \left(\dot{\theta}_1 + \dot{\theta}_2\right) \mathrm{s}_{12} \end{bmatrix}}_{j(\theta)} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix},$$

where $J(\theta)$ is the Jacobian for velocities expressed as $(\dot{x}, \dot{y})$. If we chose to fix the robot's end-effector in x-axis, where the canvas is located, we can take any constant value for x, like $x = 2$. Then this holonomic constraint can be expressed in joint space θ as $L_1 \mathrm{c}_1 + L_2 \mathrm{c}_{12} = 2$, and its time derivative can be written $A(\theta)\dot{\theta} = 0$, i.e.,

$$\underbrace{\begin{bmatrix} [-L_1 \mathrm{s}_1 - L_2 \mathrm{s}_{12} & -L_2 \mathrm{s}_{12}] \end{bmatrix}}_{A(\theta)} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

For our 2RR manipulator we have 2 joint coordinates, $n = 2$ and one holonomic constraint $k = 1$ constraint, so $A(\theta) \in \mathbb{R}^{1\times 2}$. The time derivative of $A(\theta)$ is

$$\dot{A}(\theta) = \left[-L_1 \dot{\theta}_1 c_1 - L_2 \left(\dot{\theta}_1 + \dot{\theta}_2\right) c_{12} \quad - L_2 \left(\dot{\theta}_1 + \dot{\theta}_2\right) c_{12}\right].$$

3.2 Brush Dynamics

We chose to model the painting brush as a snubber mechanical system. Brush will be deformed during painting process and the more it is pressed against the canvas the more it will resist it. We thought that snubber mechanical system reflects the elasticity and deflection properties of brush tip hairs well [20]. Figure 3.4 below shows a painting brush and what it's tip would look like if we model it as a snubber system.

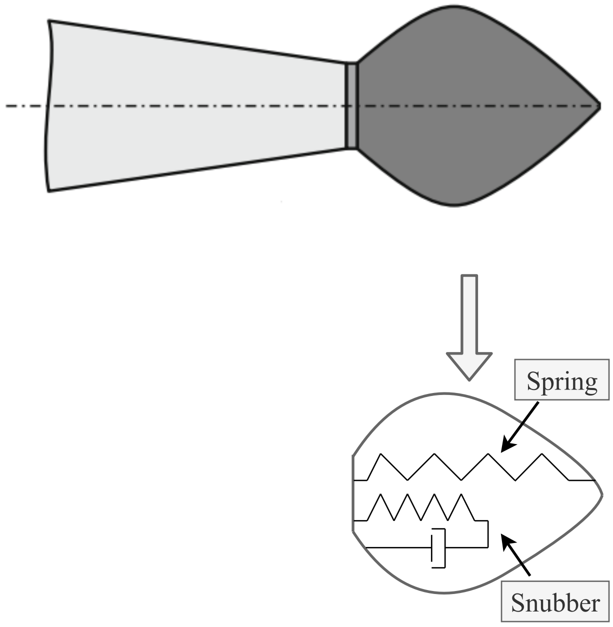

Fig. 3.4: Modeling of paintbrush as snubber system

Snubber system consists of two parts, one is the spring part where spring runs from beginning of the brush to it's tip, depending on the design choice. Other part is snubber part, which consists of spring and damper. Snubber starts around the halfway from the brush's tip. For convenience, when we talk about the whole brush tip, we will call it snubber system. But when we talk about the part with spring and damper, we will call it snubber.

In the Figure 3.5, we can observe the painting robotic system with painting brush modeled as snubber system attached to it's end-effector. We will assume that brush will be deformed after it makes contact with the canvas, meaning that canvas cannot be punched through. After the contact, it can only slide across the y-axis, painting the straight brushstroke. Another assumption will be that the brush tip will make a perpendicular contact with the canvas.

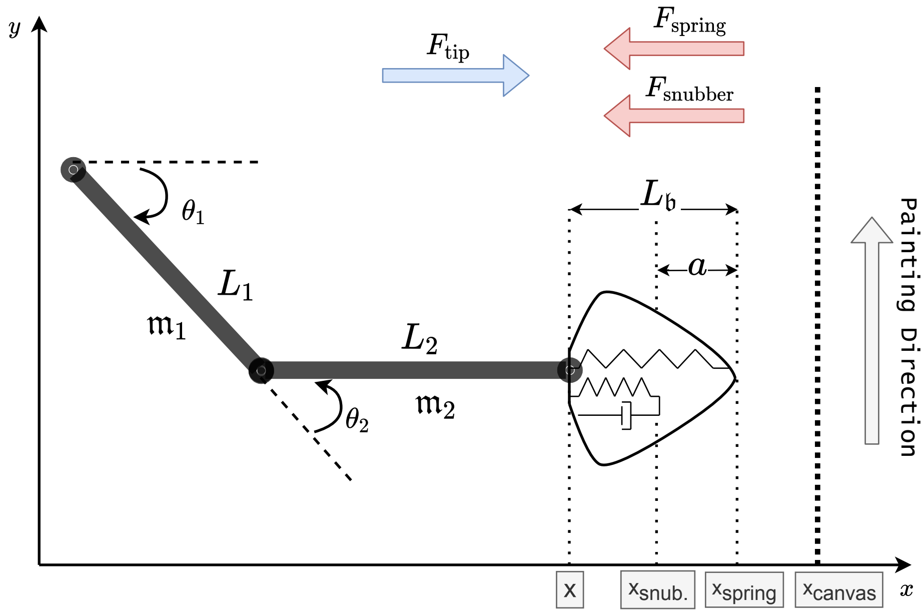

Fig. 3.5: Painting robotic system with brush attached to it's end effector

When brush tip makes contact with the canvas, F_{spring} is activated, resisting the robot's end-effector force, F_{tip} . We will define the total brush length as $L_{\mathfrak{b}}$ and the distance between the start of the snubber part and the spring as a. $\mathrm{x}_{\text{canvas}}$ stands for the position of canvas and x describes the position of end-effector in the x-axis. When brush is deformed till the snubber part, F_{Snubber} , which is made of spring and damper, gets activated. We can express $\mathrm{x}_{\text{spring}}$ and $\mathrm{x}_{\text{snubber}}$ as

$$\mathrm{x}_{\text{spring}} = x + L_{\mathfrak{b}}$$

$$\mathrm{x}_{\text{snubber}} = x + L_{\mathfrak{b}} - a$$

Then we can define the F_{Spring} and F_{Snubber} (spring and damper) as following

$$F_{\text{spring}} = \begin{cases} 0 & \text{for } x_{\text{spring}} < x_{\text{canvas}} \\ k_1\,(x + L_{\mathfrak{b}} - x_{\text{canvas}}) & \text{for } x_{\text{spring}} \geq x_{\text{canvas}} \end{cases} \tag{3.20}$$

$$F_{\text{snubber}} = \begin{cases} 0 & \text{for } x_{\text{snubber}} < x_{\text{canvas}} \\ k_2\,(x + L_{\mathfrak{b}} - a - x_{\text{canvas}}) + b_2\dot{x} & \text{for } x_{\text{snubber}} \geq x_{\text{canvas}} \end{cases} \tag{3.21}$$

where k_1 is the spring coefficient, k_2 & b_2 are spring and damper coefficients for the snubber part, respectively, and $\dot{x}$ is the velocity along the x-axis.

Dynamical equation of interaction between the brush and robot's end-effector force in x-direction, F_{tip} can be defined as

$$F_{\text{tip}} - F_{\text{spring}} - F_{\text{snubber}} = m\ddot{x} \tag{3.22}$$

If we re-write the Equation (3.22) explicitly, we get

$$\begin{gathered} F_{\text{tip}} - k_1 x - k_1 L_{\mathfrak{b}} + k_1 x_c - k_2 x - k_2 L_{\mathfrak{b}} + k_2 a + k_2 x_{\text{canvas}} - b_2 \dot{x} = m\ddot{x} \\ F_{\text{tip}} + (k_1 + k_2)(x_{\text{canvas}} - L_{\mathfrak{b}}) + k_2 a = m\ddot{x} + b_2 \dot{x} + (k_1 + k_2) x \end{gathered} \tag{3.23}$$

3.3 Friction

Model of friction is central to any model of contact manipulation and our robotic painting system is not an exception. During the sliding phase of the painting process, there is a friction between the paintbrush and the canvas. Since we assume that sliding will be only on the y-axis, we define the friction to be in opposite direction to painting on this axis. Thus we formulate our friction force as following

$$F_{\text{friction}} = \sigma |v_y| \tag{3.24}$$

where σ is viscous friction coefficient and $|v_y|$ is the absolute value of velocity along the y-axis (Fig. 3.6).

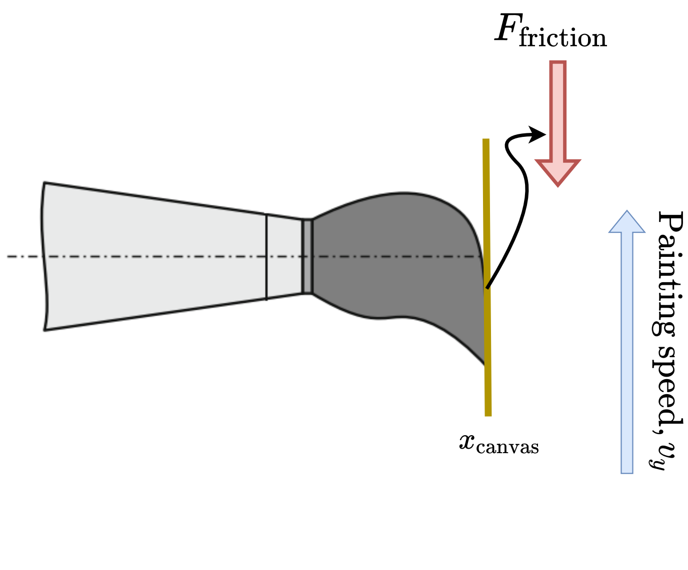

Fig. 3.6: Friction and brush tip inflection

We will assume that when contact is made, brush tip will have maximum amount of painting. It will deposit it's paint onto the canvas during sliding phase and eventually all the paint will be gone, making the brush tip dry. This dryness or wetness of the brush tip will affect the friction force. In order to reflect this, we will first introduce weighted function, g(v), for paint volume.

$$g(v) = 1 - (\text{Volume}/\text{Volume}_{\text{max}}) \tag{3.25}$$

where Volume is the variable showing paint amount on the brush, while $\text{Volume}_{\text{max}}$ is the initial amount of paint, which is also the maximum by design. At the start of the painting, g(v) takes value of 1, and when the tip of the brush is dry it takes the value of 0. We incorporate the g(v) to the viscous friction coefficient σ

$$\sigma = (\mu - \alpha)g(v) + \alpha \tag{3.26}$$

where μ is the dry friction coefficient and α is the minimum friction coefficient that is present between canvas and brush when brush paint amount is maximum. By modeling the viscous friction coefficient, σ, dependent from the paint volume amount, we assume to reflect the reality as close as possible.

3.4 Paint Transfer

In the previous section, we talked about how paint volume on the brush tip would decrease by sliding across the canvas. In this section, we will try to model this process. Paint deposit rate from brush to canvas will depend on the size of the brush, especially the brush width and how hard it is pressed against the canvas. Since we assume that the brush will be deformed when pressed against canvas, brush area that is in contact with the canvas will show a positive correlation with pressing. Harder the brush is pressed against the canvas, the more area of brush will be in contact with the canvas, thus paint deposition rate will increase as well. Figure 3.7 illustrates relation between deformation, Δx and brush tip inflection, Δz.

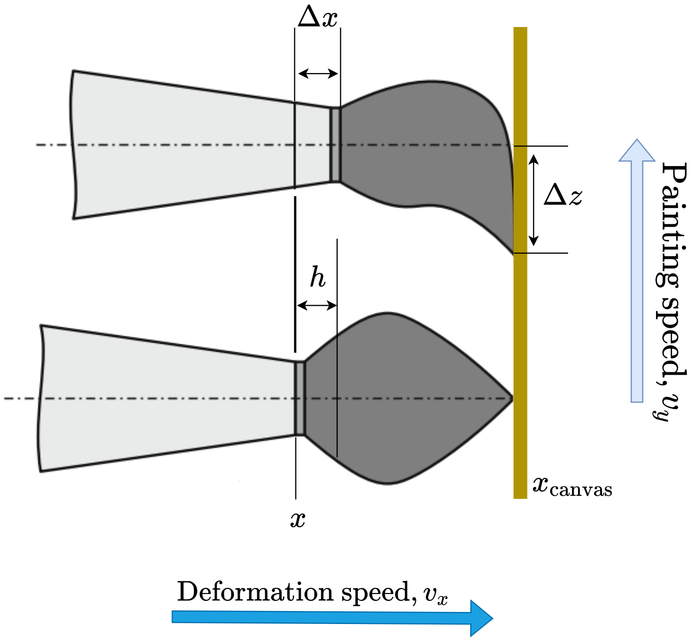

Fig. 3.7: Brush tip inflection

Thus, before proceeding with paint deposition rate, we will now define a new term, stroke width (s_w), that reflects the brush tip inflection. We will define stroke width as following

$$\mathfrak{s}_w = \mathfrak{b}_w z \tag{3.27}$$

where b_w is brush width and z is

$$z = (z_{\text{max}} - z_{\text{min}})g(\delta) + z_{\text{min}} \tag{3.28}$$

where g(δ) is weighted function for the brush inflection. z_{min} will get the value of 1 as we assume stroke width to be always great or equal than brush width. z_{max} will determined by design choice, how big we assume will be the stroke width compared to brush width when completely pressed against the canvas. g(δ) is the function that depends on maximum length that brush can be deformed and brush length,

$$g(\delta) = (\text{DeformationLength}/\text{BrushLength}) \tag{3.29}$$

Another factor that we should take into account while formulating the paint deposition rate is the paint volume amount at the tip of the brush. The dryer the brush gets, the harder it will be to get the paint flow from brush to canvas. To include this factor, we will define a new variable, β, that will relate the amount of paint volume left in the brush tip to paint deposition rate. β is defined as

$$\beta = (\beta_{\min} - \beta_{\max})g(v) + \beta_{\max} \tag{3.30}$$

where $\beta_{\max}$ value is 1 and $\beta_{\min}$ value is the percent of the $\beta_{\max}$ value, that can take values between 0 and 1.

Considering all the factors stated above, we can now define the paint deposition rate as follows:

$$\dot{V} = \mathrm{s_w}\eta\beta \tag{3.31}$$

where where $\mathrm{s_w}$ is stroke width, β relates paint volume in the brush tip to the deposition rate. η is the absorption coefficient that changes depending on the canvas material.

4 Simulation Set-up

After proposing the dynamical compliant model for the Van Gogh painting robot, we verify the model through numerical simulations. Simulation setup will be discussed in this chapter. We introduce the simulation setup, type of input parameters and the simulation settings. The numerical simulation is conducted in Matlab/Simulink. We setup the simulation in a way that it reflects the three painting phases (approach, contact and sliding) for our robotic system that we introduced in Chapter 3. Figure 4.1 describes the general overview for simulation setup. Jacobian matrix is used to make the transition from task space to joint space. During the approach phase, only robot's own dynamics are active. Constrained (cons.) dynamics, brush dynamics, and friction model get activated only after the contact is made with the canvas.

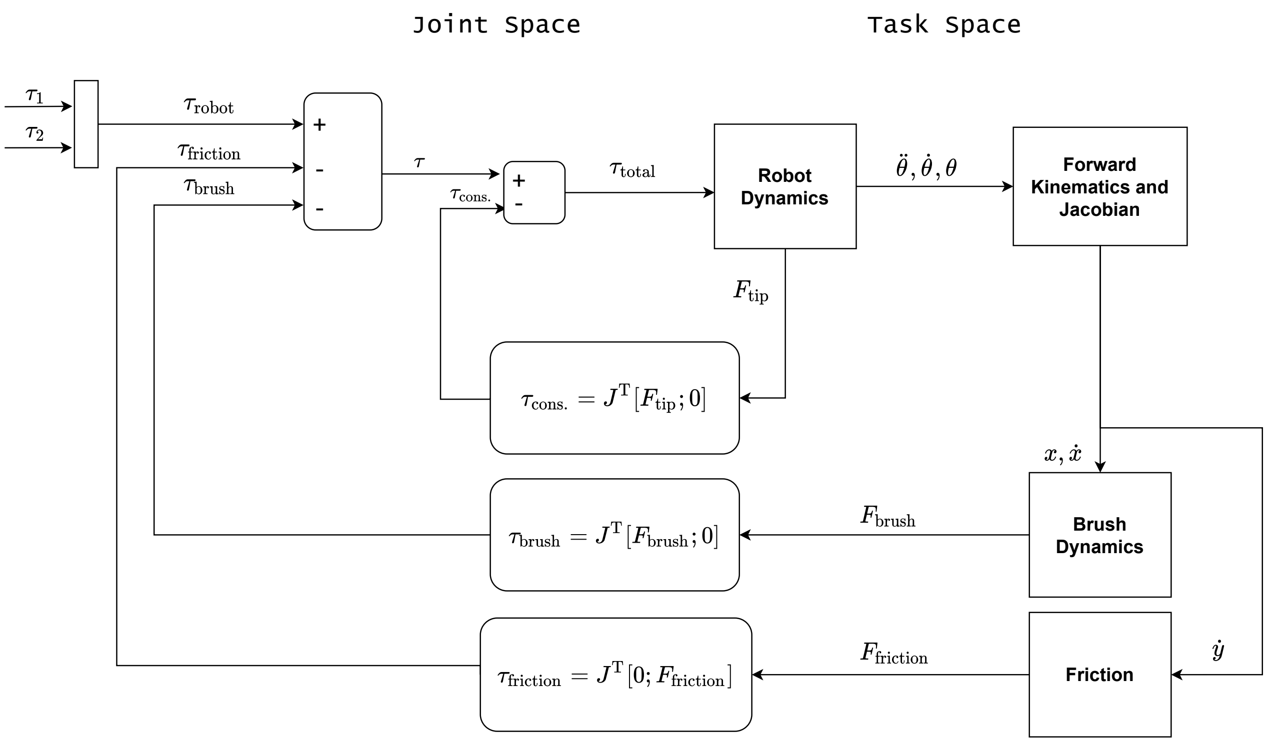

Fig. 4.1: Simulation setup overview

$\mathcal{F}_{\text{tip}}$ denotes the end-effector's force in task space. It is derived by using the Lagrange multiplier from Eq. (3.17). We use τ_{total} from Figure 4.1 to derive the Lagrange multiplier. $\tau_{\text{cons.}}$ is then calculated and feed back to the dynamical model only after the contact is made with brush's hard point (See Figure 4.2). F_{brush} and $\mathcal{F}_{\text{tip}}$ concern dynamics related to x-axis, while F_{friction} affects the dynamics in the y-axis. That is reflected in the Figure 4.1 above and explains the difference between the calculation of τ_{brush} and τ_{friction}.

4.1 Design Parameters

We introduce all the relevant design parameters in this section. Design parameters is divided into main 5 categories:

- Brush Size: We define three brush sizes, which are long, medium and short, depending on the brush hair dimensions. SilverBrush Black Velvet 3025S watercolour brushes are used as reference [21]. See Table 4.1
- Brush Type: Types of brush define how stiff the brush is. k_1 , k_2 and b_2 are the spring constant, snubber spring constant and snubber damper constant respectively. During the painting process, brush tip can be deformed till the hard part of the brush, and it is denoted by h (Fig. 4.2). a is the measurement of where the snubber starts. See Table 4.2

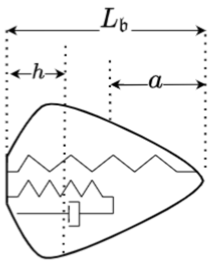

Fig. 4.2: Brush tip hard point

- Paint-Canvas: In this category we define three parameters, α, μ and η. α is the minimum value for the friction force coefficient when the amount of paint is at it's highest value. μ is the dry friction force coefficient. η is the absorption coefficient [Eq.(3.26)]. See Table 4.3
- Robot Power: It is the input power for the robot actuators. τ_1 is the actuator torque for arm 1 and τ_2 is the torque for arm 2. Together with

damping coefficient, these parameters will define the approach speed, painting speed and deformation speed.

- Robot Type: We chose the Quanser 2-DOF serial link manipulator as the 2RR serial robotic arm for our Van Gogh painting robot. Properties for this specific manipulator can be seen in Table 4.5. Generic robot arm is the values that we initialized the simulation with.

Brush Size				
Type	**Length** (mm)	**Width** (mm)	**Thickness** (mm)	**Capacity** (mL)
Long	50	16	16	12.8
Medium	42	13	13	7.1
Short	30	8	8	1.9

Tab. 4.1: Brush Size types

Brush Type					
Type/Parameters	k_1	k_2	b_2	**h**	**a**
Hard	0.4	0.5	1	L/5	L/2
Medium	0.3	0.4	1	L/5	L/2
Soft	0.2	0.3	1	L/5	L/2

Tab. 4.2: Brush stiffness types

Canvas-Paint			
Parameters	α	μ	η
Range	0.05 — 0.2	0.1 — 0.5	0.1 — 0.6

Tab. 4.3: Canvas-Paint parameters

Robot Input			
Parameters	τ_1	τ_2	**damping coeff.**
Range	1 — 5	1 — 5	0.5 — 1

Tab. 4.4: Input parameters for robotic arm

Manipulator Properties						
Type/Parameters	a_1(m)	a_2(m)	m_1(kg)	m_2(kg)	Io_1(kgm^2)	Io_2(kgm^2)
Quanser 2-DOF	0.343	0.267	1.51	0.873	0.0392	0.00808
Generic Robot	1	1	1	1	0	0

Tab. 4.5: Robotic arm properties

4.2 Outputs

Relevant parameters to observe during the simulations runs are determined to be brushstroke length, total painting time, average painting speed, paint consumption and paint quality. We start measuring brushstroke length when the contact is made between the brush and canvas till brush tip is dry. Paint consumption is the measurement of how much area is painted per mL of paint. Paint quality (φ) is the measurement of paint amount per stroke length and it's unit is mL/cm. Table 4.6 gives a summary of important outputs.

Outputs						
	total time	stroke length	average painting speed	paint area	paint consumption	paint quality
unit	s	cm	cm/s	cm2	cm2/mL	mL/cm

Tab. 4.6: Output summary

Results

5

5.1 Determining Parameters

5.1.1 Robot Input parameters

For the initial simulation run, we choose the Quanser 2-DOF manipulator arm settings, which can been in Table 4.5. As for the brush type attached to the end-effector, we selected the large brush size (Table 4.1) and hard brush type (Table 4.2). Robot input parameters, τ_1, τ_2 and damping coefficient are chosen to be 3.0, 2.6 and 0.9 respectively. We use this run to observe the approach phase, determine the approach speed and check if the constrained dynamics work as planned.

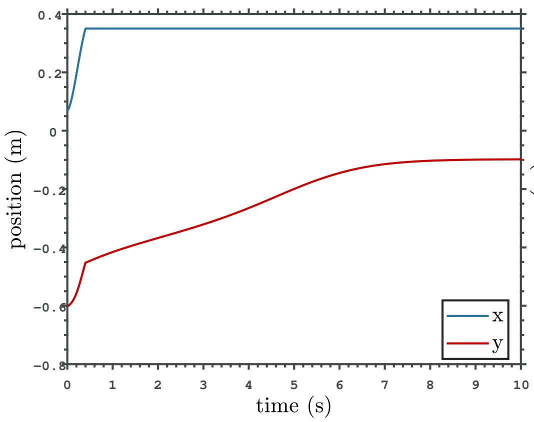

Fig. 5.1: Initial Run - End Effector Position

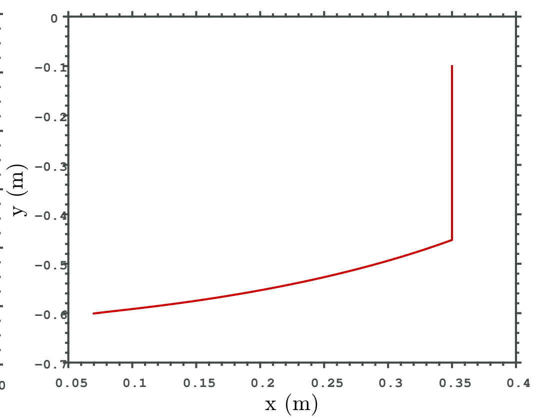

Fig. 5.2: Initial Run - End Effector Trajectory

Figure 5.1 and Figure 5.2 illustrate the end-effector position and trajectory respectively. We can observe that after brush was pressed till it's hard point, constrained dynamics get activated and robot's end-effector is fixed in x-axis, and then sliding in y-axis.

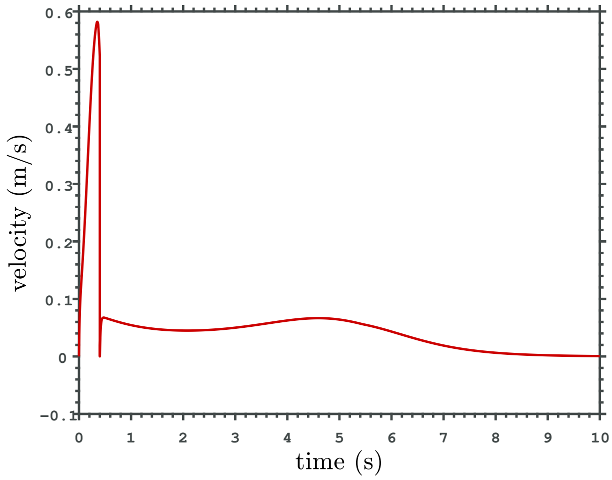

Fig. 5.3: Initial Run - End effector speed

However, when we check the approach speed in Figure 5.3 , we see that approach speed in y-direction is quite high. We try to overcome this by changing the robotic arm's initial position, bringing it closer to the canvas. Initial value for θ_1 is changed from -90°to -75°and initial value for θ_2 is changed from 15°to 30°. This setting changes the initial position along the x-axis, cutting the distance to the canvas. For the first scenario the distance was 29cm but for this run it is now 8cm. We can see the effect of these changes on the approach speed in Figure 5.4.

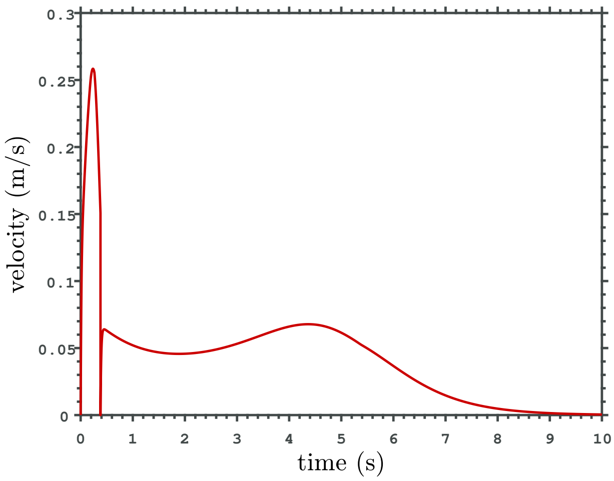

Fig. 5.4: Second Run - End effector speed

Interaction forces, F_{brush} and F_{friction} can be observed in Fig. 5.5. F_{brush} is made of two parts, F_{spring} and F_{snubber} and we can observe that damping part of snubber clearly in the graph. After the brush is completely pressed till it's hard point, velocity along the x-axis becomes zero, making the damper part of the snubber zero as well. During the painting, contact is maintained along the x-axis and velocity is equal to 0 along this axis, thus F_{brush} is constant. F_{friction} is dependent from the velocity along the y-axis (painting speed) and the viscous friction coefficient, σ. We will address this in details later in this chapter.

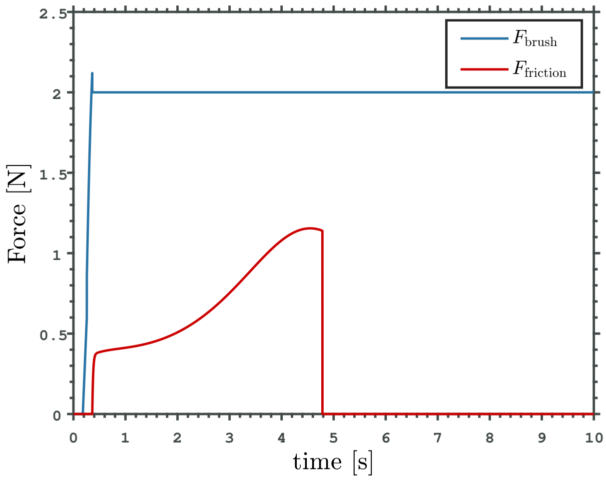

Fig. 5.5: S2 - Brush force and Friction force

5.1.2 Spring and Damper Coefficients

Spring and Damper coefficients of the brush model, k_1, k_2 and b_2 determine the magnitude of F_{brush}. These are the only parameters defining the F_{brush} magnitude because by design, brush contradiction and start of the snubber system is fixed for this set of simulation runs. After multiple runs with different values for k_1, k_2 and b_2, we determine that $k_1 = 0.4$, $k_2 = 0.5$ and $b_2 = 1$ work best for the $F_{\text{brush}} = 2$ N. If we try to increase the values of coefficients, under the current robot-input parameters, brush is not pressed fully to it's hard point. This results in not maintaining the contact with the canvas because by design our constraint force gets activated when brush is fully pressed till it's hard point. To give an example, when we increase the k_2 value from 0.5 to 0.6, brush is not compressed till it's hard point, thus the contact is not maintained, see the Figures 5.6 and 5.7. The results are the same when we increase the values for k_1 and b_2.

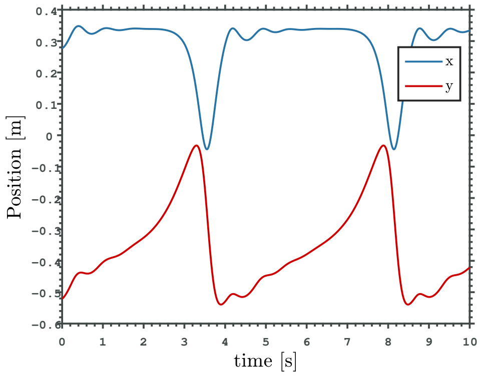

Fig. 5.6: S2 - End Effector Position

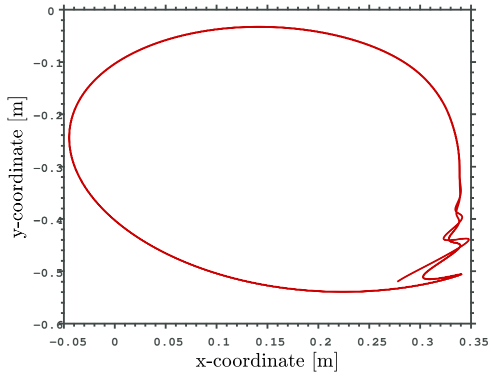

Fig. 5.7: S2 - End Effector Trajectory

5.1.3 Absorption Coefficient

Absorption coefficient, η, determines the paint deposition rate, which is the rate of paint transfer from brush to canvas. From Figures 5.8 and 5.9 we can observe that with the increasing value for η, deposition rate increases, paint volume decreases quicker. In our model, viscous friction force coefficient is dependent from the paint volume amount, thus change in absorption coefficient value will also have an effect on the friction force. We can see that in Fig. 5.10 and 5.11. Figure 5.12 shows how the painting speed varies for different values of η.

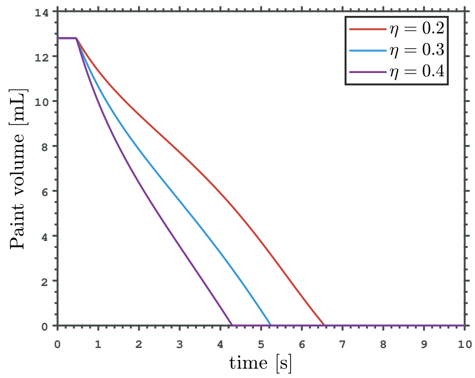

Fig. 5.8: S3 - Paint volume

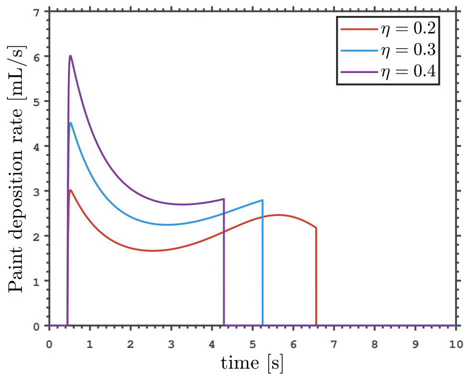

Fig. 5.9: S3 - Paint deposition rate

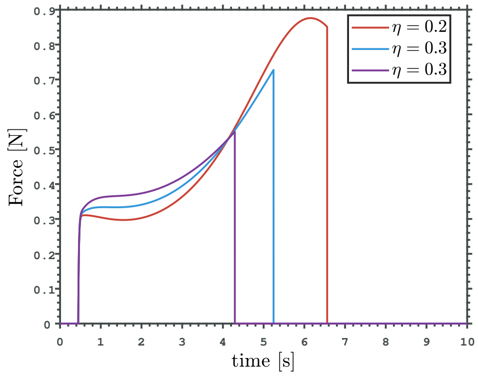

Fig. 5.10: S3 - Friction force

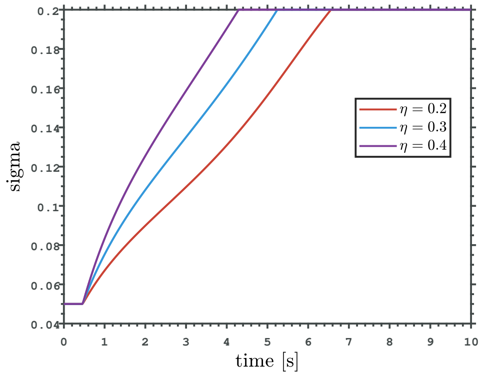

Fig. 5.11: S3 - Viscous coefficient

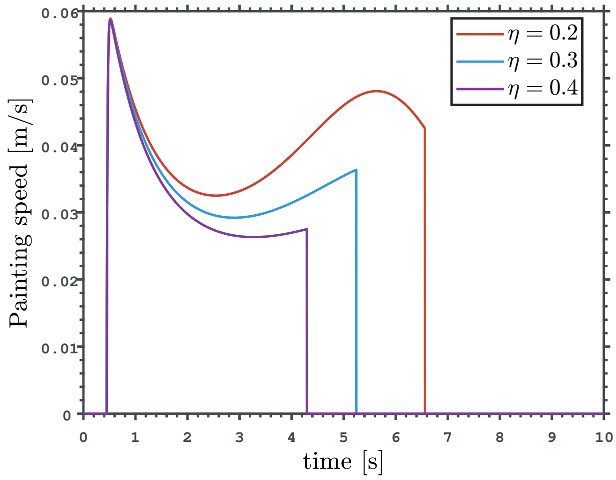

Fig. 5.12: S3 - Painting Speed

Absorption coefficient will differ from material to material that is chosen for the painting, like canvas, paper or ceramic. Table 5.1 shows the results of increasing absorption coefficient on stroke length and paint consumption. Higher the absorption coefficient shorter the stroke length, which results in smaller paint area. Given the initial amount of volume, stroke length of 25cm makes more sense than the other options, thus the absorption coefficient value will be chosen as 0.2.

Outputs					
absorption coefficient	total time (s)	stroke length (cm)	average paint speed (cm/s)	paint area (cm2)	paint consumption (cm2/mL)
0.2	5.98	25.00	4.17	52.00	4.06
0.3	4.68	16.67	3.55	34.66	2.70
0.4	3.75	12.50	3.33	26.00	2.03

Tab. 5.1: Sensitivity analysis for absorption coefficient

5.1.4 Deformation Speed

After determining the values and range for robot inputs, spring-snubber coefficients and absorption coefficient, we modify our simulation model in a way that, after the contact with the canvas, brush gets deformed slowly till it's hard point, imitating the real painting process. We can only achieve this by adding a controller for deformation speed (Fig. 5.13). Controller will make sure that brush is deformed at a constant speed along the x-axis.

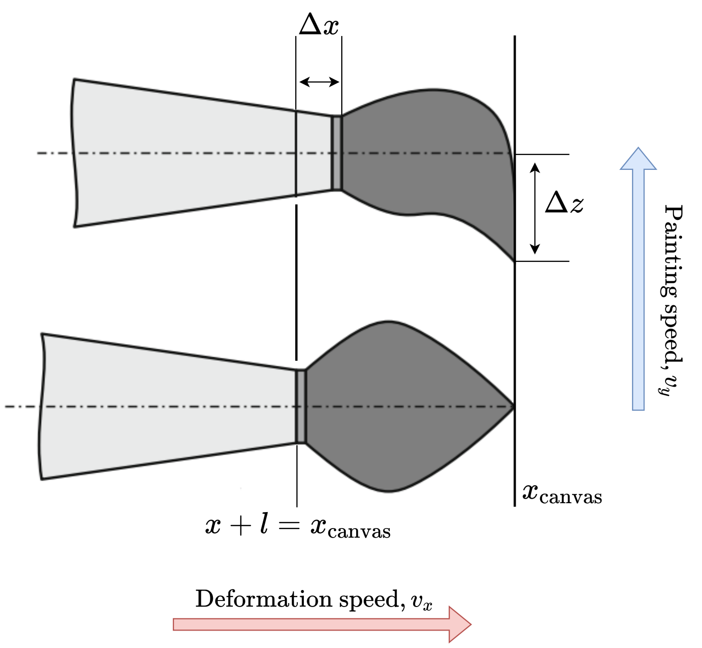

Fig. 5.13: Inflection of the brush tip

Figure 5.14 illustrates the general schematic how controller for deformation is incorporated into the existing dynamical model. We use a simple PID-controller block. F_e is the error between desired deformation speed, $\dot{x}_d$, and actual speed, $\dot{x}$. Then we convert this error from task space into joint space by the help of Jacobian matrix and feed it back to the robot dynamics.

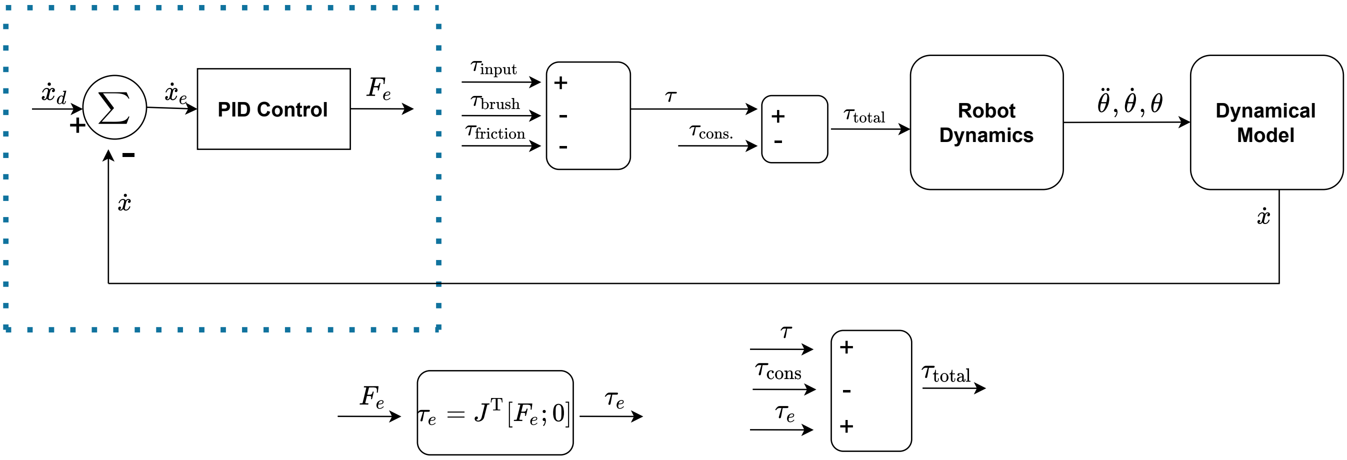

Fig. 5.14: Control schematic for deformation speed

In Figures 5.15 and 5.16, we can observe the x-trajectory of the brush and deformation speed. We chose to design controller in a way that, deformation happens till the brush hard point, in order not to punch through the canvas.

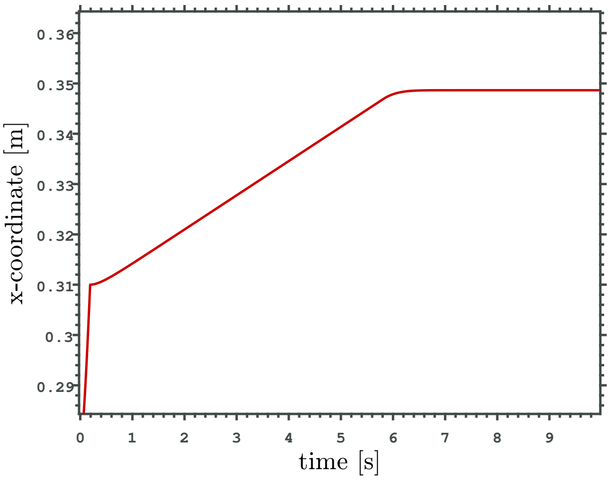

Fig. 5.15: S4 - Brush deformation reflected on x-coordinate

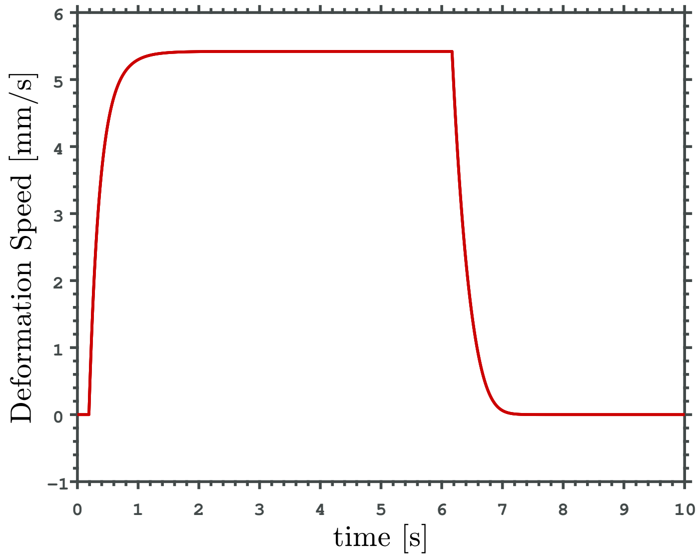

Fig. 5.16: S4 - Deformation speed

As we stated earlier in Chapter 3, modelling with brush deformation will have implications on stroke width, making it dependent on the amount of brush inflection on the canvas, see Equations (3.27), (3.28) and (3.29). Figure 5.17 shows the relation between brush width and stroke width. We chose $z_{\max}$ to be 1.5, which means if the brush is deformed to the maximum, stroke width will be equal to 1.5 times of the brush width.

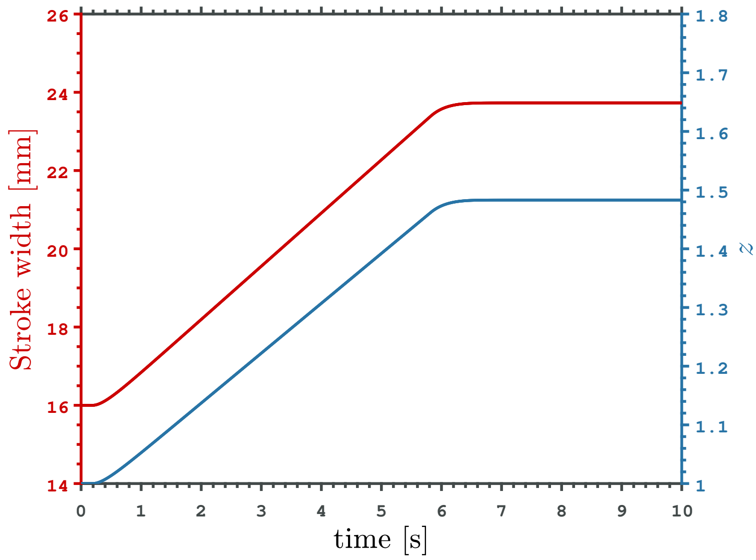

Fig. 5.17: S4 - Relation between stroke width and z

5.1.5 Paint deposition rate

As we have defined earlier in Chapter 3, paint deposition rate $\dot{V}$ was dependent on s_w (stroke width), η (absorption coefficient) and β which relates the amount of volume left in the brush tip to the deposition rate (Eq. (3.31)). β_{max} was defined as 1 and β_{min} as 0.3. It implicates that, initially when paint brush is wet, paint will flow easier to canvas than compared to when brush tip is dry. We can see the relation between paint volume amount and β in Figure 5.18. β takes values between 0.3 and 1, depending on the paint volume amount.

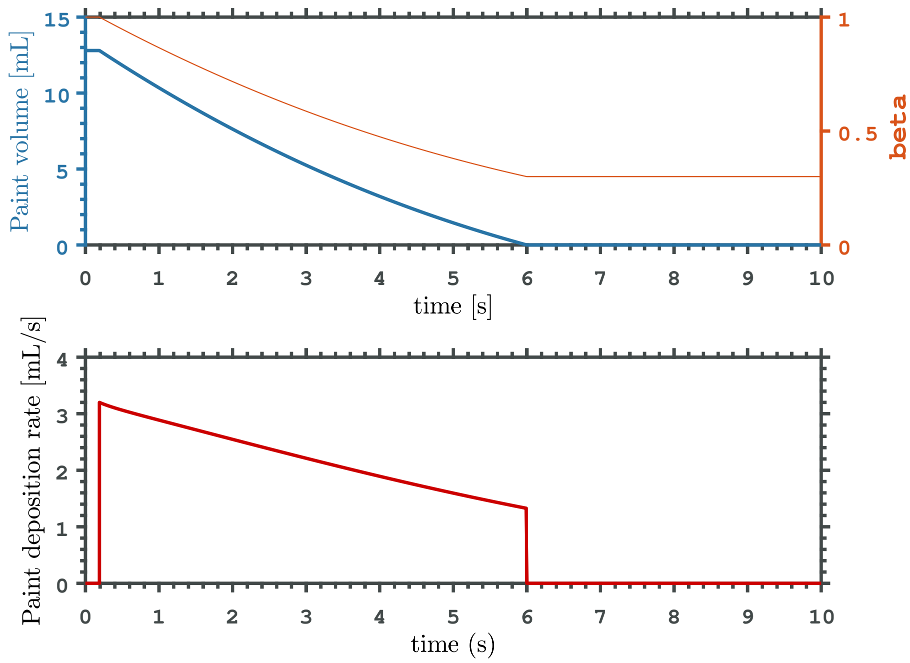

Fig. 5.18: Paint volume & deposition rate

5.1.6 Friction Coefficient

Viscous friction coefficient, σ is dependent from minimum friction coefficient, α and dry friction coefficient, μ (Eq.(3.26)). After various simulation runs, we determine the range for α as 0.05-0.1 and for μ as 0.2-0.6. In the figures below, we can see how σ, friction force and painting speed vary depending on α and μ values. We expect friction force to increase while painting, thus we choose the α as 0.05 and μ as 0.2, because it reflects this phenomena the best (Fig. 5.21).

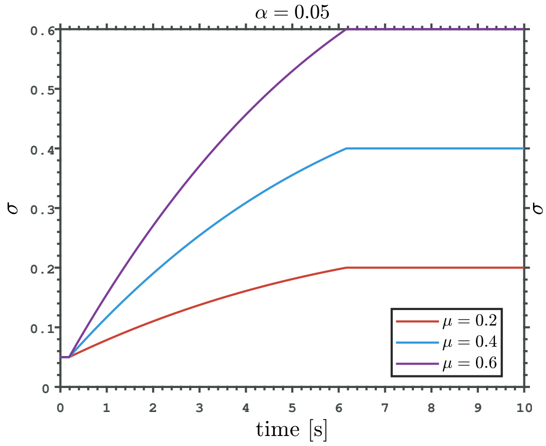

Fig. 5.19: Friction coefficient for $\alpha = 0.05$

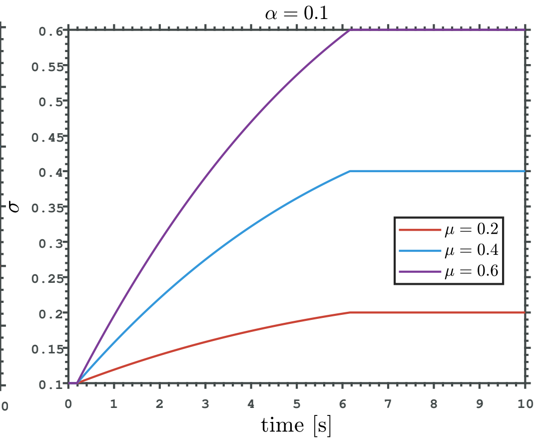

Fig. 5.20: Friction coefficient for $\alpha = 0.1$

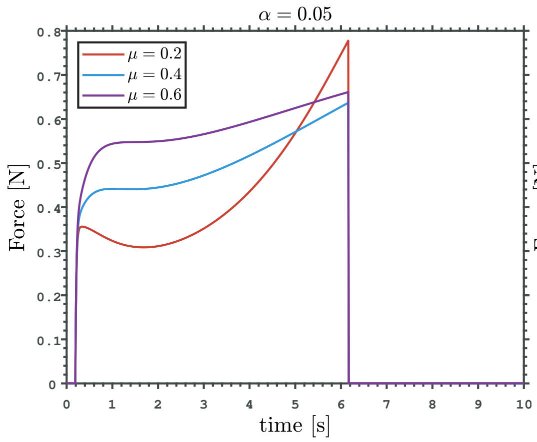

Fig. 5.21: Friction force for $\alpha = 0.05$

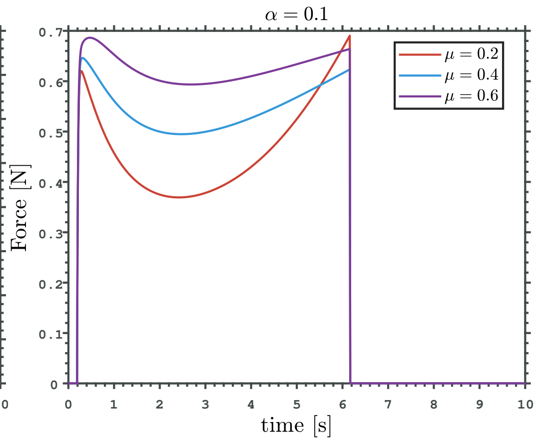

Fig. 5.22: Friction force for $\alpha = 0.1$

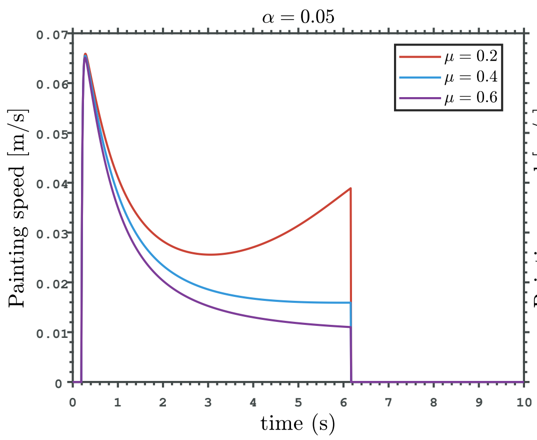

Fig. 5.23: Painting speed for $\alpha = 0.05$

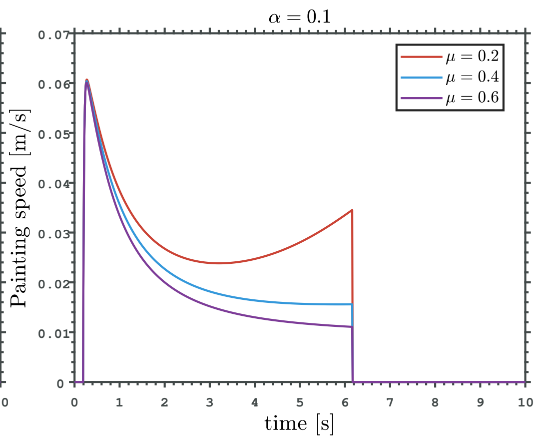

Fig. 5.24: Painting speed for $\alpha = 0.1$

5.2 Painting Speed

In the figure below, we can see how painting speed behaves during the painting process. Because it depends on robot dynamics and friction force, it varies greatly over the course of painting and this is not ideal because it affects the quality of the brushstroke. Like it was with the deformation speed, we need to incorporate a controller in order to control the painting speed (y-axis).

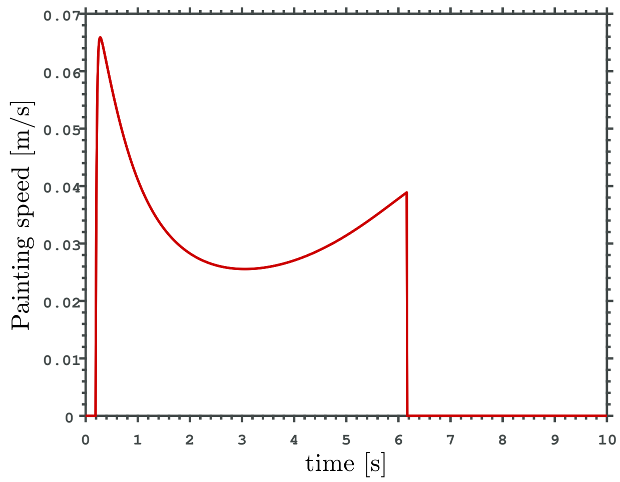

Fig. 5.25: S4 - Painting speed -without controller

Figure 5.26 describes the schematic of PID-controller for painting speed. Difference between this schematic and controller for deformation is that, error for painting speed is incorporated in τ block. The reason for this is that we want to control the speed for y-direction and have no effect in x-direction. τ block is then added to $\tau_{\text{cons.}}$ which makes sure that there is no change in x-direction.

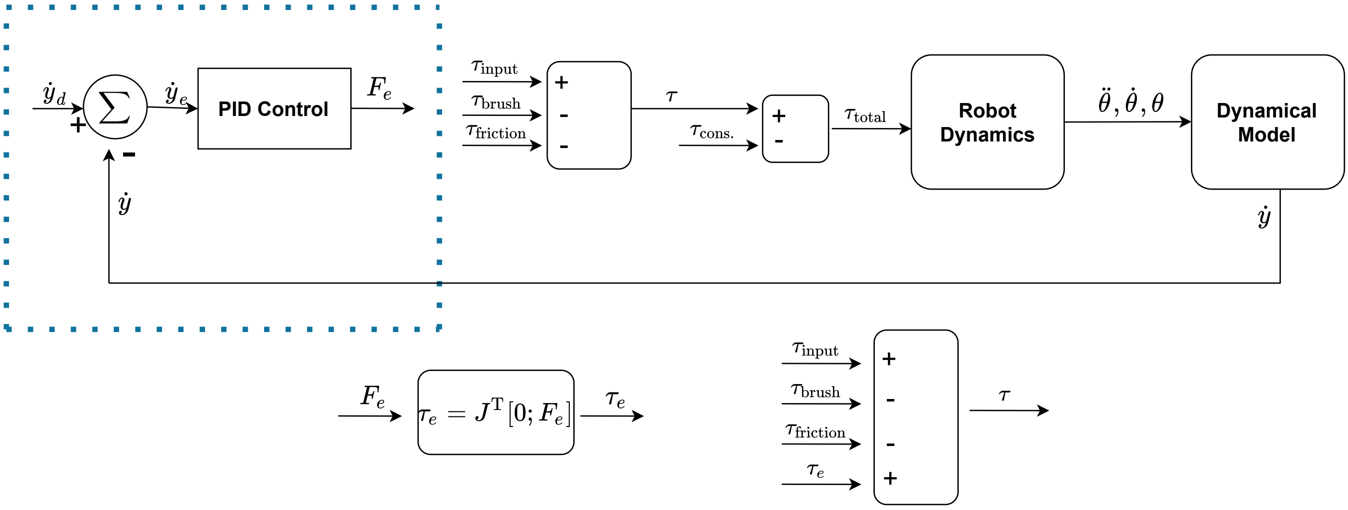

Fig. 5.26: Control schematic for painting speed

In the figure below, we can see the painting speed graph after controller is implemented. Compared to the earlier graph, painting speed is constant now. This was achieved by adjusting the value for proportional gain for PID block, K_p as 50.

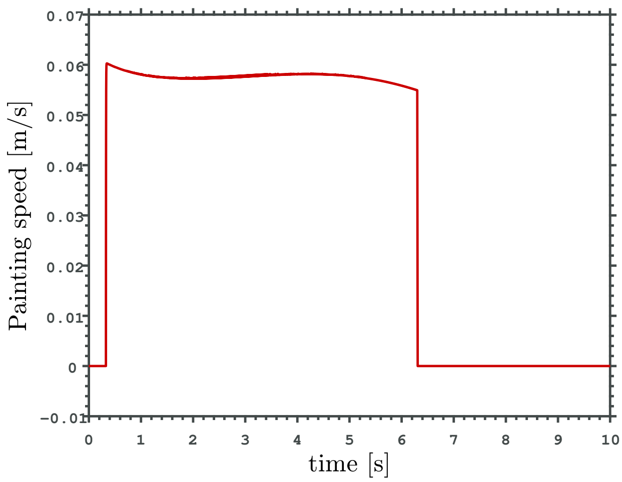

Fig. 5.27: Painting speed with controller

5.3 Brushstroke quality

After implementing controllers for both deformation and painting speed, it is time to analyze the stroke quality. We define stroke quality, φ, as an amount of paint deposited per stroke length, and defined it's unit as mL/cm. We calculate the painting quality as deposition rate divided by painting speed

$$\varphi = \frac{\dot{V}}{v_y} = \frac{s_W \eta \beta}{v_y}$$

As we can see from the formula, we can control the stroke quality through either painting speed (v_y) or indirectly through deformation speed (it will affect the stroke width parameter) or by controlling the both parameters. In the figure below, we can see that how stroke quality is changing over time. Stroke starts with 0.5mL per centimeter, and decreases all the way to 0.2mL. There is a substantial difference in quality at the start of the stroke and at the end of it even tough both deformation and painting speed is constant.

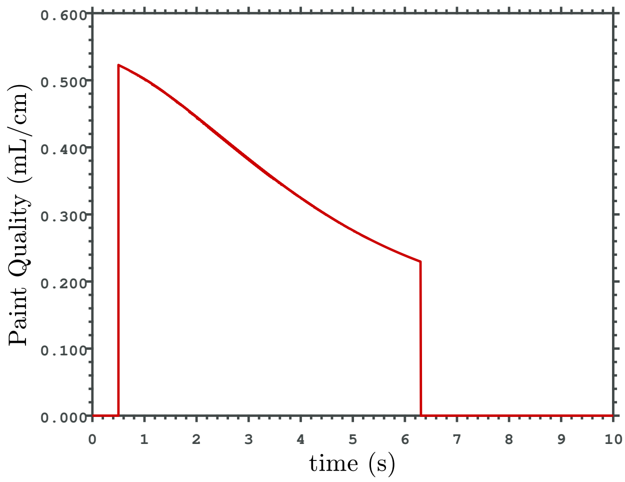

Fig. 5.28: Painting quality - no controller implemented

Thus we decide to incorporate a higher level controller for painting quality. The schematic for the proposed higher level control can be seen in the Figure 5.29. Error term for quality is positively related to the deformation speed and negatively related to the painting speed. In simpler terms, if we want to increase the painting quality, we need to deform the brush harder or/and decrease the painting speed. Proposed higher level control tries to achieve that by controlling the deformation and painting speed.

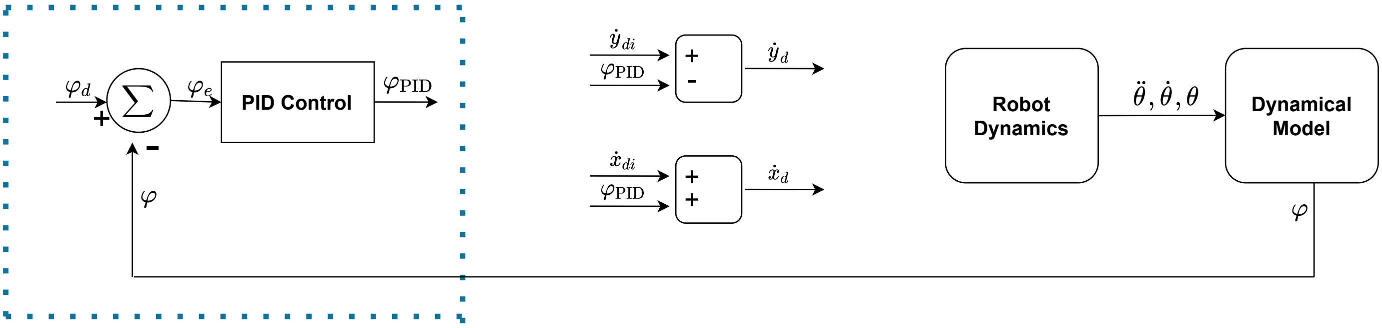

Fig. 5.29: Control schematic for painting quality

In Figure 5.30, we can see the results for painting quality with different proportional gain values set for PID-controller. When proportional gain value is set to 100, we can see that painting quality is constant throughout the stroke, thus achieving uniform, smooth brushstroke. It is achieved through controlling the painting and deformation speed, see Figure 5.31 and 5.32.

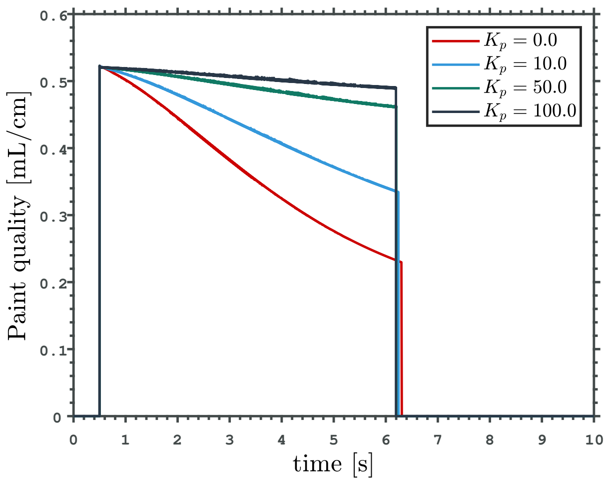

Fig. 5.30: Painting quality with different proportional gains

To keep the stroke quality constant, painting speed is decreased from 6cm/s to 3cm/s, while deformation speed is increased, increasing the paint deposition rate. We can also check how keeping stroke quality constant reflects on total painting time, stroke length and other pre-determined outputs in Table 5.2. As expected, stroke length substantially decreased because paint quality is increased with the incorporation of quality controller. Average painting speed also decreased.

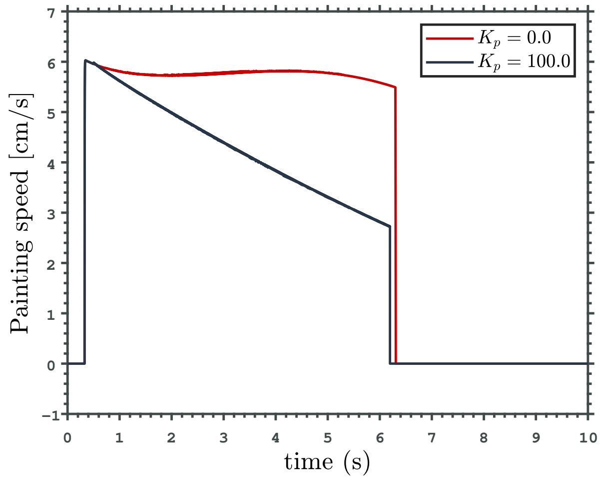

Fig. 5.31: Painting speed comparison

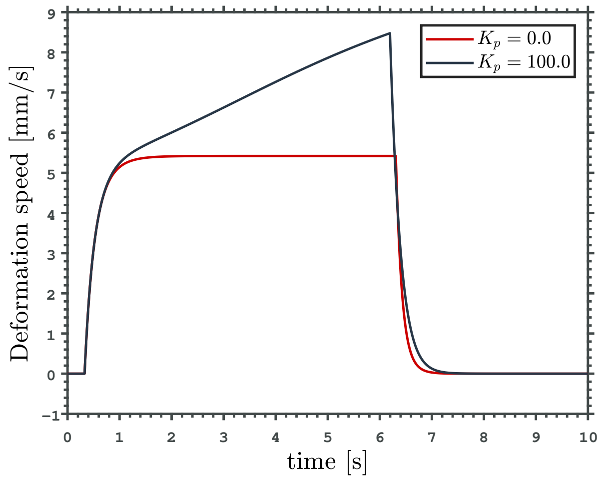

Fig. 5.32: Deformation speed comparison

Outputs					
paint quality	total time (s)	stroke length (cm)	average paint speed (cm/s)	paint area (cm2)	paint consumption (cm2/mL)
control (-)	5.98	34.42	5.75	71.61	5.59
control (+)	5.86	25.23	4.29	52.48	4.10

Tab. 5.2: Output comparison of higher level control

6 Conclusion and Discussion

In this chapter, we discuss the objective of the research, how the research was conducted, what results did we get and what are the implications of those results. Later on we will discuss the limitations of this project and present a list of suggestions to improve the current work.

The objective of this research was to develop a dynamical compliant contact model for painting robotic system that would enable painting of uniform, smooth brushstroke on the canvas.

Proposed dynamical model included the compliant interaction between painting brush and the canvas as well as dynamical modeling of painting robot, brush and paint transfer model. This dynamical model was achieved by modeling paint brush as mechanical snubber system, developing a constrained dynamics for 2RR robot, and proposing a paint transfer and friction model. Having developed a dynamical compliant model that reflected every critical step in painting process, enabled us to incorporate a simple PID-controller in order to control painting and brush deformation speed. By adding a higher level control for painting quality showed promising results in terms of controlling the brushstroke quality. It was shown that by adjusting the deformation and painting speed with the help of higher level control, we can, in theory, achieve uniform brushstrokes on the canvas.

6.1 Limitations

Van Gogh robotic painting system that was proposed in this research has a fair amount of limitations. Firstly, even tough the name suggests otherwise, it was designed to paint only on one-axis. Van Gogh would not have the fame that he has today if he had painted on one-axis. Secondly, we had some assumptions which may not be the case in real life, like painting brush will always be perpendicular to the canvas, brush tip will deform in one direction and etc. Current approach also requires tip of the manipulator to be really close to the canvas at the start, which may not be ideal for real life painting system. This limitation can be overcome by adding another controller for the robotic arm during the approach phase.

6.2 Future Work

Hereby we will present the recommendation list for future projects. The next big step in developing the robotic painting system would be actually testing of the proposed model in real life. This will create opportunity to test the assumptions made in our project. A controller to adjust the approaching speed of robotic arm can be implemented. We can also increase the painting dimension from one axis to two axis. Another recommendation would be adding another degree of freedom to the robot, like controlling the orientation of the end-effector, thus creating opportunities to paint in an artistic way.

Bibliography

[1] H. Cohen, „The further exploits of aaron, painter," *Stanford Hum. Rev.*, vol. 4, no. 2, pp. 141–158, Jul. 1995 (cit. on p. 1).

[2] P. A. Tresset and F. F. Leymarie, „Portrait drawing by paul the robot," *Comput. Graph.*, vol. 37, pp. 348–363, 2013 (cit. on p. 1).

[3] O. Deussen, T. Lindemeier, S. Pirk, and M. Tautzenberger, „Feedback-guided stroke placement for a painting machine," in *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, ser. CAe '12, Annecy, France: Eurographics Association, 2012, pp. 25–33 (cit. on p. 1).

[4] L. Scalera, S. Seriani, A. Gasparetto, and P. Gallina, „Watercolour robotic painting: A novel automatic system for artistic rendering," *Journal of Intelligent & Robotic Systems*, vol. 95, 2019. DOI: 10.1007/s10846-018-0937-y (cit. on p. 1).

[5] Robot Art. „The robotic art competition." [Accessed 10-Jan-2022]. (), [Online]. Available: https://robotart.org/ (cit. on p. 1).

[6] L. Scalera, S. Seriani, A. Gasparetto, and P. Gallina, „Artistic painting with a collaborative robot," Dec. 2020. DOI: 10.5281/zenodo.4781437 (cit. on p. 1).

[7] M. Vukobratovic, V. Potkonjak, and V. Matijevic, *Dynamics of Robots with Contact Tasks*, ser. International Series on Microprocessor-Based and Intelligent Systems Engineering. Springer, Dordrecht, 2003 (cit. on p. 2).

[8] „Kindpng." [Accessed 10-Jan-2022]. (), [Online]. Available: https://www.kindpng.com/ (cit. on p. 3).

[9] P. Veljko, M. K. Vukobratović, and V. Matijević, „Dynamics of contact tasks in robotics. part ii—case study in dynamics of contact motion," *Mechanism and Machine Theory*, vol. 34, no. 6, pp. 943–972, 1999. DOI: https://doi.org/10.1016/S0094-114X(97)00092-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094114X9700092X (cit. on p. 6).

[10] D. Verscheure, J. Swevers, H. Bruyninckx, and J. De Schutter, „On-line identification of contact dynamics in the presence of geometric uncertainties,“ in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 851–856. DOI: 10.1109/ROBOT.2008.4543311 (cit. on p. 6).

[11] L. Baptista, J. Sousa, and J. S. da Cost, „Predictive force control of robot manipulators in nonrigid environments,“ in *Industrial Robotics: Theory, Modelling and Control*, Pro Literatur Verlag, Germany / ARS, Austria, Dec. 2006. DOI: 10.5772/5013. [Online]. Available: https://doi.org/10.5772%2F5013 (cit. on pp. 7, 8).

[12] D. Marhefka and D. Orin, „Simulation of contact using a nonlinear damping model,“ in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, 1996, 1662–1668 vol.2. DOI: 10.1109/ROBOT.1996.506951 (cit. on p. 7).

[13] B. Yao and M. Tomizuka, „Adaptive Control of Robot Manipulators in Constrained Motion—Controller Design,“ *Journal of Dynamic Systems, Measurement, and Control*, vol. 117, no. 3, pp. 320–328, Sep. 1995. DOI: 10.1115/1.2799122. eprint: https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/117/3/320/5552358/320_1.pdf. [Online]. Available: https://doi.org/10.1115/1.2799122 (cit. on p. 7).

[14] M. Otsuki, K. Sugihara, A. Toda, F. Shibata, and A. Kimura, „A brush device with visual and haptic feedback for virtual painting of 3D virtual objects,“ *Virtual Reality*, vol. 22, pp. 167–181, 2018. DOI: 10.1007/s10055-017-0317-0 (cit. on pp. 8, 9).

[15] W. V. Baxter, V. Scheib, M. C. Lin, and D. Manocha, „DAB: Interactive haptic painting with 3D virtual brushes,“ in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, E. Fiume, Ed., New York, NY: ACM, 2001, pp. 461–468. DOI: 10.1145/383259.383313 (cit. on p. 9).

[16] C. Guo, Z. Hou, G. Yang, and S. Zheng, „The simulation of the brush stroke based on force feedback technology,“ *Mathematical Problems in Engineering*, vol. 2015, pp. 1–10, 2015. DOI: 10.1155/2015/164821 (cit. on pp. 10, 11).

[17] L. Scalera, S. Seriani, A. Gasparetto, and P. Gallina, „Non-photorealistic rendering techniques for artistic robotic painting,“ *Robotics*, vol. 8, no. 1, 2019. DOI: 10.3390/robotics8010010. [Online]. Available: https://www.mdpi.com/2218-6581/8/1/10 (cit. on p. 12).

[18] K. Lynch and F. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017. [Online]. Available: https://books.google.nl/books?id=0B4xDwAAQBAJ (cit. on pp. 18, 22).

[19] A. J. Koivo, *Fundamentals for Control of Robotic Manipulators*, 1st. New-York: John-Wiley, 1989 (cit. on p. 21).

[20] B. T. Kulakowski, J. F. Gardner, and J. L. Shearer, *Dynamic Modeling and Control of Engineering Systems*, 3rd ed. Cambridge University Press, 2007, pp. 157–165. DOI: 10.1017/CBO9780511805417 (cit. on p. 24).

[21] SilverBrush Limited. „3000s black velvet.“ [Accessed 25-Jan-2022]. (), [Online]. Available: https://www.silverbrush.com/3000s-black-velvetreg-series.html (cit. on p. 31).