2021 International Siberian Conference on Control and Communications (SIBCON) | 978-1-7281-8504-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/SIBCON50419.2021.9438857

Robot arm planning and control by τ-$Jerk$ theory and vision-based recurrent ANN observer

Ivan Carvajal and Edgar A. Martínez-García*
Laboratorio de Robótica, Inst. of Eng. and Technology
Universidad Autonoma de Ciudad Juárez, Mexico
0000-0001-9163-8285*

Roman Lavrenov and Evgeni Magid
Institute of Information Technologies and Intelligent Systems
Kazan Federal University, Russian Federation
0000-0003-1696-8545 and 0000-0001-7316-5664

***Abstract*—This work describes a planning path-tracking control for a 6-axis robot manipulator in palettes assembly. Two biologically inspired approaches motivated this work: the general τ-$\mathcal{J}erk$ theory for trajectory tracking and a recurrent bi-layer Hopfield artificial neural network. Equidistant Cartesian points generate free-collision paths between the robot and the palette. Nonlinear regression-based 3rd grade polynomials represents polynomial assembling trajectories. A variational method optimizes paths length. The method is validated through numeric simulations, showing feasibility and effectiveness.**

***Index Terms*—robotic-arm, assembling, model-based-control, tau-theory, vision, Hopfield-neurons, by-layer-ANN**

I. Introduction

Today's manufacturing systems and organizations strongly depend on the need to use numerous intelligent control techniques and scientific computing models to fulfill their production goals [1]. This work is motivated by the need to control a 6-axis robotic arm in automatic assembling of self-locking nylon ties in palettes of cable harnesses. The focus of this work is inspired by two biological approaches for robot's trajectory tracking control: the general τ-*Jerk* theory for trajectory control [2] [3]; and a recurrent Hopfield-based bilayer ANN as visual observer. The ANN outputs determine the target regions, from which new assembling trajectories are generated. The robot's kinematic model was deduced for tracking motion control. A red-green-blue-Depth (RGB-D) visual sensor was fixed at the scenario's top location to obtain exteroceptive views [5]. The RGB-D sensor provides both, 2D intensity-based images and metric 3D cloud points. From multiple arbitrary robot's postures, different trajectories to the palette's regions of interest are yielded. The highest priority path is optimized by minimizing two criteria: the robot-palette distance and the path's number of inflection points. In industrial manufacture, robot arms are extensively deployed, SCARA-type robots [6], peg-in-hole assembly tasks [7], [8], assembling for haptic-device-based trajectory teaching and control [9], components assembly on continuously-moving lines [12] or multi-robot with multi-visual systems for automotive front wheel components assembly [10]. Moreover, vision-based deep-learning techniques [8] and research progress on neural networks for controlling manipulators [11] for industrial applications are becoming popularly used. Further, the τ theory establishes that biological entities yield goal-directed motions from perceptual reflects depending on the world's geometrical inter-relations. In [13], an optimal trajectory planning technique of robot arms involving kinematic motion constraints and values of velocity, acceleration and Jerk was presented. The work [14] reported a method to obtain optimal time-Jerk robot motion applying 5^o B-spline interpolations to construct a path. Similarly, [15] presented a meta-heuristic optimal path planner of a welding robot. This paper is organized by the following sections. Section II presents the kinematic model. Section III describes the τ-Jerk model for path generation. Section IV indtroduces the bilayer neural visual observer. Section V presents the path optimization method. Finally, section VI provides conclusions.

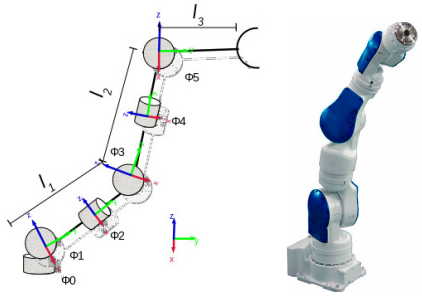

Fig. 1. Kinematic diagram of 6-axis robotic arm (SIA20).

II. Robotic arm kinematics

The robot kinematics (showed in Fig.1) is deduced to its 1^{st}-order linear matrix form. Let us provide the definition II.1:

Definition II.1. Let c_0, c_{01}, s_2 represent a short notation of functions $\cos(\phi_0), \cos(\phi_0+\phi_1)$ and $\sin(\phi_2)$, respectively.

Based on Definition II.1, the Cartesian model for robot of Fig.1 is expressed by (1)-(3),

$$x = l_1 c_1 s_0 + l_2(c_{13}s_0c_2 + s_{13}s_2) + l_3[c_{135}s_0c_2c_4 + s_{135}c_2s_4 - c_{135}s_0s_2s_4 + s_{135}s_2c_4], \quad (1)$$

for the y-component,

$$y = (l_1c_1 + l_2c_{13} + l_3c_{135})c_0 \quad (2)$$

and

$$z = l_1 s_1 s_0 - l_2(c_{13}s_0s_2 - s_{13}c_2) - l_3[c_{135}s_0s_2c_4 - s_{135}s_2s_4 + c_{135}s_0c_2s_4 - s_{135}c_2c_4]. \quad (3)$$

It follows from previous position expressions that the first-order derivatives describing the robot arm's Cartesian speeds $\dot{\mathbf{p}}$ is given in terms of the joints rotary speeds $\dot{\mathbf{\Phi}}$. Thus, it can be expressed in the linear matrix form as $\dot{\mathbf{p}} = \mathbf{J} \cdot \dot{\mathbf{\Phi}}$. The matrix $\mathbf{J}$ is non stationary time-varying Jacobian matrix. The Jacobian elements are deduced next by the functions partial derivatives w.r.t. joints angular variables. Thus, the derivatives for the function x are:

$$\frac{\partial x}{\partial \phi_0} = l_1 \mathrm{c}_1 \mathrm{c}_0 + l_2(\mathrm{c}_{13}\mathrm{c}_0\mathrm{c}_2) + l_3(\mathrm{c}_{135}\mathrm{c}_0\mathrm{c}_2\mathrm{c}_4 - \mathrm{c}_{135}\mathrm{c}_0\mathrm{s}_2\mathrm{s}_4), \tag{4}$$

$$\frac{\partial x}{\partial \phi_1} = -l_1 \mathrm{s}_1 \mathrm{s}_0 + l_2(-\mathrm{s}_{13}\mathrm{s}_0\mathrm{c}_2 + \mathrm{c}_{13}\mathrm{s}_2) + l_3(-\mathrm{s}_{135}\mathrm{s}_0\mathrm{c}_2\mathrm{c}_4 + \mathrm{c}_{135}\mathrm{c}_2\mathrm{s}_4 + \mathrm{s}_{135}\mathrm{s}_0\mathrm{s}_2\mathrm{s}_4 + \mathrm{c}_{135}\mathrm{s}_2\mathrm{c}_4), \tag{5}$$

$$\frac{\partial x}{\partial \phi_2} = -l_2(\mathrm{c}_{13}\mathrm{s}_0\mathrm{s}_2 + \mathrm{s}_{13}\mathrm{c}_2) - l_3(\mathrm{c}_{135}\mathrm{s}_0\mathrm{s}_2\mathrm{c}_4 + \mathrm{s}_{135}\mathrm{s}_2\mathrm{s}_4 + \mathrm{c}_{135}\mathrm{c}_0\mathrm{s}_2\mathrm{s}_4 - \mathrm{s}_{135}\mathrm{c}_2\mathrm{c}_4), \tag{6}$$

$$\frac{\partial x}{\partial \phi_3} = -l_2(\mathrm{s}_{13}\mathrm{s}_0\mathrm{c}_2 - \mathrm{c}_{13}\mathrm{s}_2) - l_3(\mathrm{s}_{135}\mathrm{s}_0\mathrm{c}_2\mathrm{c}_4 - \mathrm{c}_{135}\mathrm{c}_2\mathrm{s}_4 - \mathrm{s}_{135}\mathrm{s}_0\mathrm{s}_2\mathrm{s}_4 - \mathrm{c}_{135}\mathrm{s}_2\mathrm{c}_4), \tag{7}$$

$$\frac{\partial x}{\partial \phi_4} = -l_3(\mathrm{c}_{135}\mathrm{s}_0\mathrm{c}_2\mathrm{s}_4 - \mathrm{s}_{135}\mathrm{c}_2\mathrm{c}_4 + \mathrm{c}_{135}\mathrm{s}_0\mathrm{s}_2\mathrm{c}_4 + \mathrm{s}_{135}\mathrm{s}_2\mathrm{s}_4), \tag{8}$$

$$\frac{\partial x}{\partial \phi_5} = l_3(-\mathrm{s}_{135}\mathrm{s}_0\mathrm{c}_2\mathrm{c}_4 + \mathrm{c}_{135}\mathrm{c}_2\mathrm{s}_4 + \mathrm{s}_{135}\mathrm{s}_0\mathrm{s}_2\mathrm{s}_4 + \mathrm{c}_{135}\mathrm{s}_2\mathrm{c}_4). \tag{9}$$

Likewise, the partial derivatives of y w.r.t. joints angular variables are given as:

$$\frac{\partial y}{\partial \phi_0} = -(l_1\mathrm{c}_1 + l_2\mathrm{c}_{13} + l_3\mathrm{c}_{135})\mathrm{s}_0, \tag{10}$$

$$\frac{\partial y}{\partial \phi_1} = (-l_1\mathrm{s}_1 - l_2\mathrm{s}_{13} - l_3\mathrm{s}_{135})\mathrm{c}_0, \tag{11}$$

$$\frac{\partial y}{\partial \phi_2} = 0, \tag{12}$$

$$\frac{\partial y}{\partial \phi_3} = (-l_2\mathrm{s}_{13} - l_3\mathrm{s}_{135})\mathrm{c}_0, \tag{13}$$

$$\frac{\partial y}{\partial \phi_4} = 0, \tag{14}$$

$$\frac{\partial y}{\partial \phi_5} = -l_3\mathrm{s}_{135}\mathrm{c}_0. \tag{15}$$

Similarly, the partial derivatives of z w.r.t. joints angular variables are

$$\frac{\partial z}{\partial \phi_0} = l_1\mathrm{s}_1\mathrm{c}_0 - l_2(\mathrm{c}_{13}\mathrm{c}_0\mathrm{s}_2) - l_3(\mathrm{c}_{135}\mathrm{c}_0\mathrm{s}_2\mathrm{c}_4 + \mathrm{c}_{135}\mathrm{c}_0\mathrm{c}_2\mathrm{s}_4), \tag{16}$$

$$\frac{\partial z}{\partial \phi_1} = l_1\mathrm{c}_1\mathrm{s}_0 + l_2(\mathrm{s}_{13}\mathrm{s}_0\mathrm{s}_2 + \mathrm{c}_{13}\mathrm{c}_2) + l_3(\mathrm{s}_{135}\mathrm{s}_0\mathrm{s}_2\mathrm{c}_4 + \mathrm{c}_{135}\mathrm{s}_2\mathrm{s}_4 + \mathrm{s}_{135}\mathrm{s}_0\mathrm{c}_2\mathrm{s}_4 + \mathrm{c}_{135}\mathrm{c}_2\mathrm{c}_4), \tag{17}$$

$$\frac{\partial z}{\partial \phi_2} = l_1\mathrm{s}_1\mathrm{s}_0 - l_2(\mathrm{c}_{13}\mathrm{s}_0\mathrm{s}_2 - \mathrm{s}_{13}\mathrm{c}_2) - l_3[\mathrm{c}_{135}\mathrm{s}_0\mathrm{s}_2\mathrm{c}_4 - \mathrm{s}_{135}\mathrm{s}_2\mathrm{s}_4 + \mathrm{c}_{135}\mathrm{s}_0\mathrm{c}_2\mathrm{s}_4 - \mathrm{s}_{135}\mathrm{c}_2\mathrm{c}_4], \tag{18}$$

$$\frac{\partial z}{\partial \phi_3} = l_2(-\mathrm{s}_{13}\mathrm{s}_0\mathrm{s}_2 - \mathrm{c}_{13}\mathrm{c}_2) - l_3(-\mathrm{s}_{135}\mathrm{s}_0\mathrm{s}_2\mathrm{c}_4 - \mathrm{c}_{135}\mathrm{s}_2\mathrm{s}_4 - \mathrm{s}_{135}\mathrm{s}_0\mathrm{c}_2\mathrm{s}_4 - \mathrm{c}_{135}\mathrm{c}_2\mathrm{c}_4), \tag{19}$$

$$\frac{\partial z}{\partial \phi_4} = l_3[-\mathrm{c}_{135}\mathrm{s}_0\mathrm{s}_2\mathrm{s}_4 - \mathrm{s}_{135}\mathrm{s}_2\mathrm{c}_4 + \mathrm{c}_{135}\mathrm{s}_0\mathrm{c}_2\mathrm{c}_4 + \mathrm{s}_{135}\mathrm{c}_2\mathrm{s}_4] \tag{20}$$

and

$$\frac{\partial z}{\partial \phi_5} = l_3(-\mathrm{s}_{135}\mathrm{s}_0\mathrm{s}_2\mathrm{c}_4 - \mathrm{c}_{135}\mathrm{s}_2\mathrm{s}_4 - \mathrm{s}_{135}\mathrm{s}_0\mathrm{c}_2\mathrm{s}_4 - \mathrm{c}_{135}\mathrm{c}_2\mathrm{c}_4). \tag{21}$$

Furthermore, the robot's end-effectors pose (Euler orientation) is modeled by its pitch $\alpha = \phi_1 + \phi_3 + \phi_5$, roll $\beta = \phi_2 + \phi_4$ and yaw $\gamma = \phi_0$ angles. The *Jacobian* works as a controllability matrix that keeps time-varying converging error by successive approximations in control loop, as given by Proposition II.1.

Proposition II.1. *The kinematic model describes the geometric constraints in terms of its independent control rotary joints:*

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \phi_0} & \frac{\partial x}{\partial \phi_1} & \frac{\partial x}{\partial \phi_2} & \frac{\partial x}{\partial \phi_3} & \frac{\partial x}{\partial \phi_4} & \frac{\partial x}{\partial \phi_5} \\ \frac{\partial y}{\partial \phi_0} & \frac{\partial x}{\partial \phi_1} & 0 & \frac{\partial x}{\partial \phi_3} & 0 & \frac{\partial x}{\partial \phi_5} \\ \frac{\partial z}{\partial \phi_0} & \frac{\partial x}{\partial \phi_1} & \frac{\partial x}{\partial \phi_2} & \frac{\partial x}{\partial \phi_3} & \frac{\partial x}{\partial \phi_4} & \frac{\partial x}{\partial \phi_5} \\ 0 & \frac{\partial \alpha}{\partial \phi_1} & 0 & \frac{\partial \alpha}{\partial \phi_3} & 0 & \frac{\partial \alpha}{\partial \phi_5} \\ 0 & 0 & \frac{\partial \beta}{\partial \phi_2} & 0 & \frac{\partial \beta}{\partial \phi_4} & 0 \\ \frac{\partial \gamma}{\partial \phi_0} & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \dot{\phi_0} \\ \dot{\phi_1} \\ \dot{\phi_2} \\ \dot{\phi_3} \\ \dot{\phi_4} \\ \dot{\phi_5} \end{pmatrix}. \tag{22}$$

The errors are feedback terms that converge by the *Jacobian*, decreasing overtime. From previous deductions, the general 1st-order derivative kinematics was analytically obtained as (22), which will take a relevant roll in the $\tau - \mathcal{J}erk$ formulation subsequently.

III. τ-$\mathcal{J}erk$ TRAJECTORY GENERATION

The bioinspired τ theory works for any perceptual system, particularly when interacting with close objects. All control motion tasks are mainly purposed to reduce the spatial separation through a motion-gap. A system has a τ model related to a type of gap, which might be represented by a distance, angle, force, etc [17]. A motion-gap is a spatial distance with remaining time $\tau(t)$ between the actual motion in progression and the target being reached, thus

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{x}{\tau}, \tag{23}$$

where speed $\mathrm{d}x/\mathrm{d}t$ of a current motion-gap is located at gap x. By dropping-off τ and rearranging the form of state variables,

$$\tau(t) = \frac{x(t)}{dx/dt} = \frac{x(t)}{\dot{x}(t)}. \tag{24}$$

In Cartesian acceleration, a robotic arm begins and ends in zero, hence, the acceleration can be coupled to the position

gap at the end of the motion. According to the general τ theory, the τ-gap is defined as the expression (24). By coupling the τ-gap through expression (25), two or more objective functions can be coupled allowing them to close all their gaps simultaneously. The coupled equation for gap τ_q depends on another gap τ_p with constant parameter $k_{q,p}$ that is proportional to displacement $x(t)$ for gap q and p,

$$\tau_q = k_{q,p} \cdot \tau_p, \tag{25}$$

and for the gap q, it is defined by

$$q = C \cdot p^{1/k_{q,p}}. \tag{26}$$

Hence, $q_0 = x_0$ and p_0 compound the total period of time $\mathcal{T}$ to close the *Jerk*-gap. Thus, following previous expression, the constant value C is defined as

$$C = \frac{q_0}{p_0^{1/k_{p,q}}}, \tag{27}$$

where q_0 is the initial and final state difference of the gap q. Further, p_0 is the initial and final states difference of gap p,

$$C = \frac{q_0}{p_0^{1/k_{p,q}}} p^{1/k_{qp}} \tag{28}$$

From coupling equation (25) the gap x is coupled with an intrinsic gap produced by a $\mathcal{J}erk$ (over acceleration) $[m/s^3]$,

$$x_{\mathcal{J}} = \frac{1}{6}\mathcal{J}\mathcal{T}_d^3 - \frac{1}{6}\mathcal{J}t^3, \tag{29}$$

the first-order derivative is obtained as a solution of the *jerking* effect displacement $x_{\mathcal{J}}$ by

$$\dot{x}_{\mathcal{J}} = \frac{1}{2}\mathcal{J}t^2. \tag{30}$$

In addition, the 2^{nd}-order derivative as function of time is

$$\tau_{\mathcal{J}}(t) = \frac{x}{\dot{x}} = \frac{1}{3}\left(t - \frac{\mathcal{T}_d^3}{t^2}\right), \tag{31}$$

when $p_0 = \mathcal{T}^3$ and $p = \mathcal{T}^3 - t^3$ and substituting into equation (28), the following expression comes up,

$$x(t) = \frac{x_0}{\mathcal{T}^{3/k_{p,q}}}\left(\mathcal{T}^3 - t^3\right)^{1/k_{qp}}, \tag{32}$$

where (32) is the robotic arm Cartesian position (Figure 2). The spatial position $\mathbf{p} \in \mathbb{R}^3$ such that $\mathbf{p} = (x, y, z)^\top$ and $p_d = \|\mathbf{p}\|$ and $p(t) = \|\mathbf{p}(t)\|$. Hence, to find a distance gap $\triangle p = p_d - p(t)$, which is substituted in

$$p(t) = \frac{\triangle p}{\mathcal{T}^{3/k_{p,q}}}\left(\mathcal{T}^3 - t^3\right)^{1/k_{qp}}. \tag{33}$$

Therefore, for the inverse solution (angles ϕ_i), the gap is stated in terms of angles $\triangle\phi = \phi_d - \phi(t)$, thus

$$\phi(t) = \frac{\triangle \phi}{\mathcal{T}^{3/k_{p,q}}}\left(\mathcal{T}^3 - t^3\right)^{1/k_{qp}}. \tag{34}$$

The τ-*Jerk* method does not directly/inversely relate the angles with Cartesian positions of the kinematic system, but the gap-specific variable is determined instead. A mathematical method to solve the inverse of a system of non linear equations

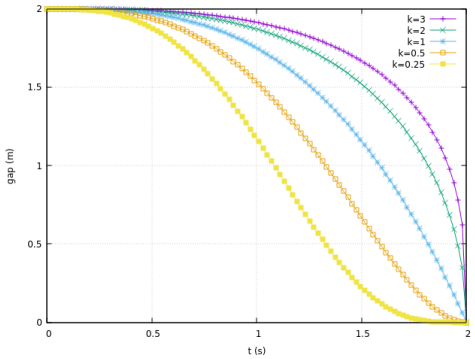

Fig. 2. τ-*Jerk at different* κ *and* $\mathcal{T} = 2.0$, $x_0 = 2.0$.

(1)-(3) and α, β, γ are required. The multivariate *Taylor* series (35) yield six equations and independent variables. To find the roots, a nonlinear system is linearized by truncating up to the 2nd term, expressed by

$$\begin{aligned}
x(\mathbf{\Phi}) &= x_t + (\phi_{0_{t+1}} - \phi_{0_t})\frac{\partial x_t}{\partial \phi_0} + \cdots + (\phi_{5_{t+1}} - \phi_{5_t})\frac{\partial x_t}{\partial \phi_5} \\
y(\mathbf{\Phi}) &= y_t + (\phi_{0_{t+1}} - \phi_{0_t})\frac{\partial y_t}{\partial \phi_0} + \cdots + (\phi_{5_{t+1}} - \phi_{5_t})\frac{\partial y_t}{\partial \phi_5} \\
z(\mathbf{\Phi}) &= z_t + (\phi_{0_{t+1}} - \phi_{0_t})\frac{\partial z_t}{\partial \phi_0} + \cdots + (\phi_{5_{t+1}} - \phi_{5_t})\frac{\partial z_t}{\partial \phi_5} \\
\alpha(\mathbf{\Phi}) &= \alpha_t + (\phi_{0_{t+1}} - \phi_{0_t})\frac{\partial \alpha_t}{\partial \phi_0} + \cdots + (\phi_{5_{t+1}} - \phi_{5_t})\frac{\partial \alpha_t}{\partial \phi_5} \\
\beta(\mathbf{\Phi}) &= \beta_t + (\phi_{0_{t+1}} - \phi_{0_t})\frac{\partial \beta_t}{\partial \phi_0} + \cdots + (\phi_{5_{t+1}} - \phi_{5_t})\frac{\partial \beta_t}{\partial \phi_5} \\
\gamma(\mathbf{\Phi}) &= \gamma_t + (\phi_{0_{t+1}} - \phi_{0_t})\frac{\partial \gamma_t}{\partial \phi_0} + \cdots + (\phi_{5_{t+1}} - \phi_{5_t})\frac{\partial \gamma_t}{\partial \phi_5}
\end{aligned} \tag{35}$$

Thus, the system is algebraically reorganized, grouping at the left-sided of equality the unknown $\mathbf{\Phi}_{t+1}$, which is the solution of interest. At right-sided, the known terms at t are

$$\begin{aligned}
\frac{\partial x_t}{\partial \phi_0}\phi_{0_{t+1}} + \cdots + \frac{\partial x_t}{\partial \phi_5}\phi_{5_{t+1}} &= -x_t + \frac{\partial x_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial x_t}{\partial \phi_5}\phi_{5_t} \\
\frac{\partial y_t}{\partial \phi_0}\phi_{0_{t+1}} + \cdots + \frac{\partial y_t}{\partial \phi_5}\phi_{5_{t+1}} &= -y_t + \frac{\partial y_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial y_t}{\partial \phi_5}\phi_{5_t} \\
\frac{\partial z_t}{\partial \phi_0}\phi_{0_{t+1}} + \cdots + \frac{\partial z_t}{\partial \phi_5}\phi_{5_{t+1}} &= -z_t + \frac{\partial z_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial z_t}{\partial \phi_5}\phi_{5_t} \\
\frac{\partial \alpha_t}{\partial \phi_0}\phi_{0_{t+1}} + \cdots + \frac{\partial \alpha_t}{\partial \phi_5}\phi_{5_{t+1}} &= -\alpha_t + \frac{\partial \alpha_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial \alpha_t}{\partial \phi_5}\phi_{5_t} \\
\frac{\partial \beta_t}{\partial \phi_0}\phi_{0_{t+1}} + \cdots + \frac{\partial \beta_t}{\partial \phi_5}\phi_{5_{t+1}} &= -\beta_t + \frac{\partial \beta_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial \beta_t}{\partial \phi_5}\phi_{5_t} \\
\frac{\partial \gamma_t}{\partial \phi_0}\phi_{0_{t+1}} + \cdots + \frac{\partial \gamma_t}{\partial \phi_5}\phi_{5_{t+1}} &= -\gamma_t + \frac{\partial \gamma_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial \gamma_t}{\partial \phi_5}\phi_{5_t}
\end{aligned} \tag{36}$$

The system is expressed by Proposition III.1,

Proposition III.1. *The inverse kinematic solution in the linear matrix form is* $\mathbf{J}_t \cdot \mathbf{\Phi}_{t+1} = \boldsymbol{\zeta}_t$, *or*

$$\begin{pmatrix} \frac{\partial x_t}{\partial \phi_0} & \frac{\partial x_t}{\partial \phi_1} & \cdots & \frac{\partial x_t}{\partial \phi_5} \\ \frac{\partial y_t}{\partial \phi_0} & \frac{\partial y_t}{\partial \phi_1} & \cdots & \frac{\partial y_t}{\partial \phi_5} \\ \frac{\partial z_t}{\partial \phi_0} & \frac{\partial z_t}{\partial \phi_1} & \cdots & \frac{\partial z_t}{\partial \phi_5} \\ \frac{\partial \alpha_t}{\partial \phi_0} & \frac{\partial \alpha_t}{\partial \phi_1} & \cdots & \frac{\partial \alpha_t}{\partial \phi_5} \\ \frac{\partial \beta_t}{\partial \phi_0} & \frac{\partial \beta_t}{\partial \phi_1} & \cdots & \frac{\partial \beta_t}{\partial \phi_5} \\ \frac{\partial \gamma_t}{\partial \phi_0} & \frac{\partial \gamma_t}{\partial \phi_1} & \cdots & \frac{\partial \gamma_t}{\partial \phi_5} \end{pmatrix} \cdot \begin{pmatrix} \phi_{0_{t+1}} \\ \phi_{1_{t+1}} \\ \phi_{2_{t+1}} \\ \phi_{3_{t+1}} \\ \phi_{4_{t+1}} \\ \phi_{5_{t+1}} \end{pmatrix}$$

$$= \begin{pmatrix} -x_t + \frac{\partial x_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial x_t}{\partial \phi_5}\phi_{5_t} \\ -y_t + \frac{\partial y_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial y_t}{\partial \phi_5}\phi_{5_t} \\ -z_t + \frac{\partial z_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial z_t}{\partial \phi_5}\phi_{5_t} \\ -\alpha_t + \frac{\partial \alpha_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial \alpha_t}{\partial \phi_5}\phi_{5_t} \\ -\beta_t + \frac{\partial \beta_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial \beta_t}{\partial \phi_5}\phi_{5_t} \\ -\gamma_t + \frac{\partial \gamma_t}{\partial \phi_0}\phi_{0_t} + \cdots + \frac{\partial \gamma_t}{\partial \phi_5}\phi_{5_t} \end{pmatrix}. \quad (37)$$

The non-stationary matrix $\mathbf{J}_t$ is the *Jacobian* as in (22). The vector prediction $\mathbf{\Phi}_{t+1} = (\phi_{0_{t+1}}, \ldots, \phi_{5_{t+1}})^\top$ is the solution to be found. Where $\boldsymbol{\zeta}_t$ is the vector of current known valued functions and angles. Thus, the inverse solution is

$$\mathbf{\Phi}_{t+1} = \mathbf{J}_t^{-1} \cdot \boldsymbol{\zeta}_t. \quad (38)$$

When all values at time t allow calculating predictions at time $t+1$, the time is updated $t = t+1$ to recursively calculate an approximated solution $\mathbf{\Phi}_{t+1}$. The best numerical fit is found when the convergence criterion $\|(\mathbf{\Phi}_{t+1} - \mathbf{\Phi}_t)/\mathbf{\Phi}_{t+1}\| < \varepsilon_\Phi$ is accomplished. The inverse solution found allows creating a gap of rotation. The τ-$\mathcal{J}erk$ requires adjustment of time $\mathcal{T}$ and coupling constant κ, obtained by setting (32). As $\mathcal{T}$ successively increases, it produced better results of the magnitude. For instance, (34) was adjusted for $\kappa = 0.4$ and $\mathcal{T} = 5$. For the angles gap of (33), the angular motion are obtained from the inverse solution. Figure 3 shows a plot of a controlled path using joints angular gap solution. In this work

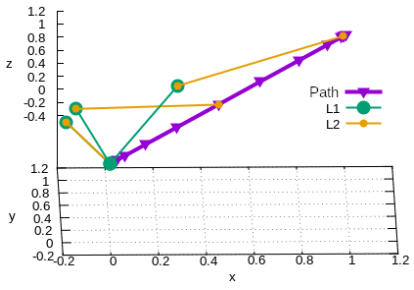

Fig. 3. *Robotic arm xyz position using* τ*au-*$\mathcal{J}erk$ *angular gaps.*

deduced the τ-$\mathcal{J}$erk theory by considering distances, angles and speeds from the gaps robot–palette.

IV. Recurrent ANN visual observer

This approach proposed an engine of propositional expressions parameterized with thresholds to separate pixel regions of the palette's elements: cables, background-base, rivets, cable ties and target assembling zones. The regions' centroid in space domain and their intensities are used as the ANN inputs. We assumed an RGB image as a cubic matrix $\mathbf{I} \in \mathbb{R}^{m \times n \times 3}$ compounded by the union of three color channels namely red $\mathbf{I}_R$, green $\mathbf{I}_G$ and blue $\mathbf{I}_B$. Definition IV.1 describes color-based segmentation criteria:

Definition IV.1. Let ζ_P be a threshold of the palette background to produce a Boolean image $\mathbf{I}_P$ by

$$\mathbf{I}_P = (\mathbf{I}_r > \zeta_P) \cap (\mathbf{I}_g > \zeta_P) \cap (\mathbf{I}_b > \zeta_P) \quad (39)$$

Let ζ_C be a threshold of black cables, resulting an image $\mathbf{I}_C$,

$$\mathbf{I}_C = (\mathbf{I}_r < \zeta_C) \cap (\mathbf{I}_g < \zeta_C) \cap (\mathbf{I}_b < \zeta_C). \quad (40)$$

Let ζ_N, ζ_{N_1} and ζ_{N_2} be the color thresholds for cables extreme connectors and image $\mathbf{I}_N$,

$$\mathbf{I}_N = ((\mathbf{I}_r < \zeta_N) \cap (\mathbf{I}_g < \zeta_N)) \cup (\zeta_{N_1} < \mathbf{I}_b < \zeta_{N_2}). \quad (41)$$

Let ζ_{T_1}, ζ_{T_2} and ζ_T be thresholds of self-locking nylon ties for image $\mathbf{I}_T$,

$$\mathbf{I}_T = (\zeta_{T_1} < \mathbf{I}_r < \zeta_{T_2}) \cup ((\mathbf{I}_g < \zeta_T) \cap (\mathbf{I}_b < \zeta_T)). \quad (42)$$

Figure 4 shows the palette's element scatter data in the RGB space. Different approaches on using ANN for robotic control

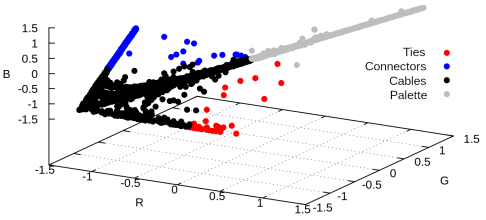

Fig. 4. *Scatter of visual data plotted in the RGB space.*

and perception have been reported [18] [19]. The Hopfield artificial neural networks (HANN) are recurrent systems, in which each neuron's output is connected to the rest of the neurons as additional synaptic weights (Figure 5). In the proposed HANN's architecture there are two layers configured as competitive neurons. The first layer is trained to detect the palette's elements of interest. The second layer works as a faults tolerant and discriminates irrelevant palette's elements. Each neuron provides a feedback signal to the rest of the same layer's neurons. The HANN has an input vector $\mathbf{x}_i(t)$ and a scalar output $y_i(t)$ binary $(0, 1)$ or polar $(-1, 1)$.

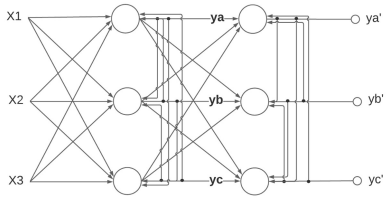

Fig. 5. *Bilayer Hopfield ANN for palette's elements recognition.*

The HANN's first layer training produces a numeric weighting matrix $\mathbf{W}_k$ per input sample. An input samples is a vector input $\mathbf{x}_k$ from the RGB space. The training rule starts by

$$\mathbf{W}_k = \sum_{i=1}^{m} \mathbf{x}_i \cdot \mathbf{x}_i^\top, \tag{43}$$

for an input sample k, $\forall\ k = \{1, 2, \ldots, m\}$, the total weights matrix $\mathbf{W}$ is optimally obtained by

$$\mathbf{W} = \mathbf{W}_1 + \mathbf{W}_2 + \cdots + \mathbf{W}_m, \tag{44}$$

the weights training matrix arises from the input $\mathbf{x}_i = (x_1, x_2, x_3, y_1, y_2)^\top$ (3 neurons and 2 feedback outputs) by

$$\mathbf{W}_{i1} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 & x_3 & y_1 & y_2 \end{pmatrix}, \tag{45}$$

expecting that $y_a = 1 \Rightarrow y_1 = y_b, y_2 = y_c$; when it's expected $y_b = 1 \Rightarrow y_1 = y_a, y_2 = y_c$ and when expected $y_c = 1 \Rightarrow y_1 = y_a, y_2 = y_b$. Therefore,

$$\mathbf{W} = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,4} & w_{1,5} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,4} & w_{2,5} \\ w_{3,1} & w_{3,2} & w_{3,3} & w_{3,4} & w_{3,5} \\ w_{4,1} & w_{4,2} & w_{4,3} & w_{4,4} & w_{4,5} \\ w_{5,1} & w_{5,2} & w_{5,3} & w_{5,4} & w_{5,5} \end{pmatrix} \tag{46}$$

Thus, a neuron's inner activation function is valued by $\mathbf{s}_i$

$$\mathbf{s}_i = \mathbf{W} \cdot \mathbf{x}_i. \tag{47}$$

A neuron's output y_i produces $[+1, -1]$. Any component s_j of the vector $\mathbf{s}_i$, $s_j < 0$ will produce an output $y_j = -1$, thus

$$y_i = \begin{cases} 1, & s_j \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{48}$$

Table I shows the following classification. The HANN outputs behave as a competitive network where only a single neuron is activated. The combinatorial inputs x_1, x_2, x_3 represent segmented image discriminated by the outputs y_1, y_2, y_3. In case of detecting false positive/negative fault, all outputs are classified negatives.

TABLE I
HOPFIELD ANN'S FIRST LAYER INPUTS AND OUTPUTS.

x_1	x_2	x_3	y_a	y_b	y_c	class
0	0	0	1	-1	-1	cables
0	0	1	-1	1	1	connector
0	1	0	-1	-1	-1	rivet
0	1	1	-1	-1	-1	–
1	0	0	-1	-1	1	tie
1	0	1	-1	-1	-1	–
1	1	0	-1	-1	-1	–
1	1	1	-1	-1	-1	palette

Moreover, the second neurons layer was trained using the same training algorithm, with differences of combinatorial input/output. Thus, the second layer weights matrix $\mathbf{W}_{i2}$ is

$$\mathbf{W}_{i2} = \begin{pmatrix} y_a \\ y_b \\ y_x \\ y_1' \\ y_2' \end{pmatrix} \cdot \begin{pmatrix} y_a & y_b & y_c & y_1' & y_2' \end{pmatrix}, \tag{49}$$

The second layer works as a faults tolerant detector outputting, the palette's elements of faults. Table II shows the following out classification:

TABLE II
HOPFIELD ANN'S SECOND LAYER INPUTS AND OUTPUTS.

y_a	y_b	y_c	y_a'	y_b'	y_c'	class
0	0	0	1	1	1	other objects
0	0	1	-1	-1	1	target (tie)
0	1	0	-1	1	-1	connector
0	1	1	-1	-1	-1	fault
1	0	0	1	-1	-1	cable
1	0	1	-1	-1	-1	fault
1	1	0	-1	-1	-1	fault
1	1	1	-1	-1	-1	fault

The 1st layer detects the palette's elements using RGB data. The 2nd layer inputs are polar outputs $[-1, 1]$ to identify relevant numeric labels by non combinatorial polar outputs $[-1, +1]$. Figure 6 shows the HANN detection results. Faults detection are due to contrast and lighting variations.

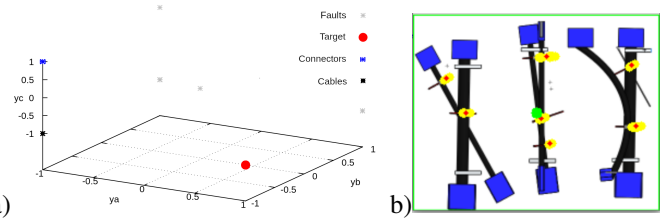

Fig. 6. *HANN-based detection. a) Output space; b) visual detection.*

V. MULTI-PATH OPTIMIZATION

Robot arm free-collision planning is a fundamental function [20]. This work optimizes paths from the robot's arbitrary posture to the palette (Figure 7). The polynomial assumes a Cartesian coordinate function $y(x)$ processing a total of n_T measurement Cartesian points. Thus, let $y(x) = a_0 + a_1 x_i + \cdots + a_n x^n$ be a measure of the Cartesian trajectory for positions in the plane XY. Thus, the 3rd degree polynomial coefficients are

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} n_T & \sum_i x_i & \sum_i x_i^2 & \sum_i x x_i^3 \\ \sum_i x_i & \sum_i x_i^2 & \sum_i x_i^3 & \sum_i x_i^4 \\ \sum_i x_i^2 & \sum_i x_i^3 & \sum_i x_i^4 & \sum_i x_i^5 \\ \sum_i x_i^3 & \sum_i x_i^4 & \sum_i x_i^5 & \sum_i x_i^6 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \sum_i y_i \\ \sum_i y_i x_i \\ \sum_i y_i x_i^2 \\ \sum_i y_i x_i^3 \end{pmatrix}. \tag{50}$$

By solving the system (50), the polynomial that estimates the y-coordinate as a function of the x-coordinate is

$$y(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3. \tag{51}$$

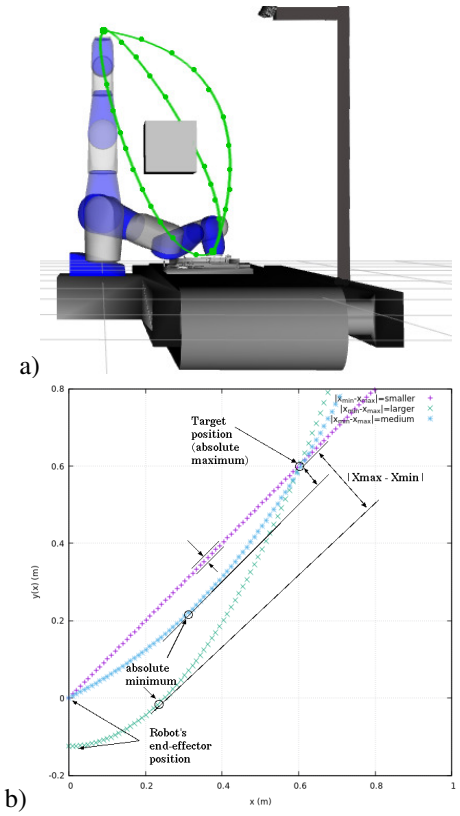

Fig. 7. *Optimized planning. a) Multi-path; b) robot's local paths.*

Therefore, (51) represents a track-path in Cartesian coordinates, which are expressed as a polynomial function $y(x)$. The 1st-order derivative of a function contains the roots that find the minimum and maximum of an original function by

$$y'(x) = a_1 + 2a_2x + 3a_3x^2, \tag{52}$$

by using the function $y'(x)$, a real root $\hat{x}_{t+1}$ is found through the Newton-Raphson root-finding method:

$$\hat{x}_{t+1} = \hat{x}_t - \frac{y'(x)}{y''(x)} = \hat{x}_t - \frac{a_1 + 2a_2x + 3a_3x^2}{2a_2 + 6a_3x}, \tag{53}$$

Substituting the root value $\hat{x}$ into the original polynomial, the valued function obtains where an optimal value exist, such that

$$y(\hat{x}) = a_0 + a_1\hat{x} + a_2\hat{x}^2 + a_3\hat{x}^3. \tag{54}$$

where $y(\hat{x})$ is an optimized value. Thus, the 2nd-order derivative is valued using the root to determine either it is a minimum or a maximum point. Thus,

$$y''(\hat{x}) = 2a_2 + 6a_3\hat{x}, \tag{55}$$

therefore,

$$\begin{cases} \max, & \text{if } y''(\hat{x}) < 0 \\ \min, & \text{if } y''(\hat{x}) > 0 \end{cases} \tag{56}$$

The absolute minimum with the shortest magnitude among all polynomials is chosen as the optimized target function, applied to either planes xy or yz separately. Figure 7b shows the robot paths in local coordinates (quasi-linear, quadratic and cubic).

VI. Conclusion

This work found out the robot's dynamic adaptability to motion from different arbitrary postures, regardless the differences of cables harness arrange, reaching assembling targets with dynamic changes of paths, despite posture and perspective changes. The results showed validity and feasibility of the proposed approach. The recurrent bi-layer ANN showed reliability for vision-based recognition of the regions of interest in assembling tasks. Finally, the assembling process is computer simulated to illustrate the robot's suitability to be deployed in real scenarios.

References

[1] A. Dogan, D. Birant, "Machine learning and data mining in manufacturing", Exp Sys with Apps, vol. 166, 2021.
[2] Y. Fang, J. Qi, J. Hu, W. Wang, Y. Peng, "An approach for jerk-continuous trajectory generation of robotic manipulators with kinematical constraints", Mech and Mach Theo, vol. 153, Nov 2020.
[3] A. Frisoli, C. Loconsole, R. Bartalucci, M. Bergamasco, "A new bounded jerk on-line trajectory planning for mimicking human movements in robot-aided neurorehabilitation", Rob and Auton Sys, 61(4), pp. 404-415, 2013.
[4] R. Song, F. Li, W. Quan, X. Yang, J. Zhao, "Skill learning for robotic assembly based on visual perspectives and force sensing", Rob and Auton Sys, vol. 135, 2021.
[5] R. Wang, A. Wu, X. Chen, J. Wang, "A point and distance constraint based 6R robot calibration method through machine vision", Robotics and Computer-Integrated Manuf, 65(10), 2020.
[6] T. Borangiu, N.A. Ivanescu, S. Barad, "Robotized Flange Assembling with Line Scan Camera Control", IFAC Proc. Volumes, 36(23), pp. 119-124, 2003.
[7] J. Jiang, Z. Huang, Z. Bi, X. Ma, G. Yu, "State-of-the-Art control strategies for robotic PiH assembly", Robotics and Computer-Integrated Manuf., vol. 65, Oct 2020.
[8] J. Song, Q. Chen, Z. Li, A peg-in-hole robot assembly system based on Gauss mixture model, Robotics and Computer-Integrated Manufacturing, vol. 67, 2021.
[9] H. Lin, "Design of an intelligent robotic precise assembly system for rapid teaching and admittance control", Robotics and Computer-Integrated Manufacturing, vol. 64, 2020.
[10] G. Collins, "Sophisticated image processing controls assembly robot", Ind. Robot, 27(6), pp.445–448, 2000.
[11] L. Jin, S. Li, J. Yu, J. He, "Robot manipulator control using neural networks: A survey" Neurocomputing, vol. 285, pp. 23-34, 2018.
[12] V. Gopinath, K. Johansen, M. Derelov, A. Gustafsson, S. Axelsson, "Safe collaborative assembly on a continuously moving line with large industrial robots, Robotics and Computer-Integrated Manufacturing, vol.67, 2021.
[13] A. Gasparetto, V. Zanotto, "A technique for time-jerk optimal planning of robot trajectories", Robotics and Computer-Integrated Manuf, 24(3), pp. 415-426, 2008.
[14] J. Huang, P. Hu, K. Wu, M. Zeng, "Optimal time-jerk trajectory planning for industrial robots", Mech and Mach Theo, 121(3), pp. 530-544, 2018.
[15] A. Rout, M. Dileep, G.B. Mohanta, B.B.V.L. Deepak, B.B. Biswal, "Optimal time-jerk trajectory planning of 6 axis welding robot using TLBO method", Proc Comp Sci, vol. 133, pp. 537-544, 2018.
[16] Z. Zhang, X. Yang, "Bio-inspired motion planning for reaching movement of a manipulator based on intrinsic tau jerk guidance", Adv. Manuf. 7, 315–325 (2019).
[17] D.N. Lee, "General tau theory: evolution to date", Perception, 38(6), 2009.
[18] S. S. Young, P. D. Scott, N. M. Nasrabadi, "Object recognition using multilayer Hopfield neural network", Ind. Robot, Apr 2020.
[19] E. Salari and S. Zhang, "Integrated recurrent neural network for image resolution enhancement from multiple image frames", in IEEE Proc. Vision, Image and Signal Proc, 150(5), pp. 299, Oct. 2003.
[20] R. C. Luo and C. Kuo, "Intelligent Seven-DoF Robot With Dynamic Obstacle Avoidance and 3-D Object Recognition for Industrial Cyber–Physical Systems" Proc. IEEE Manuf Autom., vol. 104, no. 5, pp. 1102-1113, May 2016.