Empirical and Experimental Study for Investigating Hydrodynamics of Paint Mixing Processes

Majdi Salman[a], Jasem Tamimi[a*], Maher Al-Jabari[a]

[a]Department of Mechanical Engineering, College of Engineering, Palestine Polytechnic University, P. O. Box 198, Hebron, Palestine.
*Email address: jtamimi@ppu.edu (Jasem Tamimi)

***Abstract*--** **In this paper we propose a mixing algorithm for water-based paints. This mixing algorithm guarantees user to obtain a predefined paint color within a specific time. That means, the user determines the paint volume and color as inputs to the algorithm. Then, the algorithm will specify the needed colors from red, green and blue paints to be mixed. Experimental process was done in one of the conventional mixing vessels, in which, the needed time for mixing and the time-varying color state of the mixture were determined. The color intensity of red, green and blue (RGB) distance between unmixed and mixed solution was used to specify the degree of homogeneity. This distance was obtained experimentally using a lab-made equipment. The color distance was then fitted to a suitable empirical profile. This profile helps to determine the equipment parameters such as the speed of mixing, bald shape and the required mixing time. These parameters play an important role for speed of convergence to obtain a steady homogeneous paint.**

The algorithm was applied in the MATLAB framework. For the lab equipment, we use Arduino Mega micro-controller for algorithm building and processing and high-resolution camera for paint image capturing. Furthermore, the developed system can be also used for experimental kinetic studies of various industrial and environmental problems.

***Index Term*--** **Paint mixing algorithm, paint homogeneity degree, tristimulus values, chromaticity space.**

1. Introduction

Mixing operations is a type of industrial processes which are commonly used in many industries to ensure that the final product has a certain specification. Therefore, the mixing process aims to mix the input streams to produce an outlet stream that has a desired and homogenous composition. Mixing process is an essential operation in paint industry which usually depends on automatic control techniques. The main aim of this process is to move a heterogeneous paint solution into a homogeneous one using mixing technique [1, 2]. Some of paints factories still use old equipment and tools for paint colors mixing. These equipment and tools have many problems due to processing using complicated mathematical models [3]. In order to get high quality and efficiency in paints production, recent demands of automatic paint colors preparation techniques are increased. In a batch process, raw materials are fed into the process, which then runs for a predefined period, then the final product "solution" will be available in the processing tank. In addition, no product is removed, and no additional raw materials are input, while the process is running. The main advantages of the batch reactors are that; they allow production of multiple different products in the same equipment, they are integrated enough such that they are preserved as they move from operation to operation, they are simple and very flexible as well as they are easier to clean and maintain sterile operation [4].

Obtaining a new paint with a certain color can be produced by mixing two or more paints with different colors and volumes together in a batch mixing tank [5, 6, 7]. In our batch plant, three different paint; red, green and blue, with different volumes are processed to get a new paint with a new color.

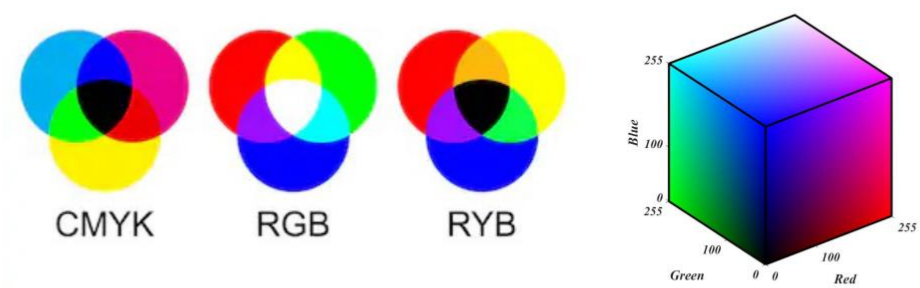
Process control of these plants has also become important in the process industries since consequences of rapidly changes in economic conditions and fast product development are highly demandable. Process control is also critical in the development of more flexible and complex processes [1, 2]. Therefore, the advancement in electronic and digital control theory will be very useful for these kinds of processes. Examples for these controllers are programmable logic controller (PLC) and supervisory control and data acquisition (SCADA) [6, 7]. Also, Arduino micro-controller can be used for the internal storage of instruction as well as implementing control functions such as logic, sequencing, timing, counting and arithmetic to control through digital, analog input or output modules of various types of these types of plants [5]. Another example of an automatic paint mixing control, also used for demonstration, is a laboratory virtual instrument engineering workbench (LabVIEW)[8]. In addition, investigating the kinetics of batch reactors and dynamic behavior of unsteady continuous stirred reaction demand an easy and a simple monitoring system for experimental studies [9, 10].

In this work, a mixing algorithm to obtain a predefined paint color is presented. Here, the user must specify the wanted paint color and volume or weight. The algorithm well then feed the suitable amounts from red, green and blue paints for the required time to reach the color homogeneity. Moreover, the algorithm is applied using a lab-made equipment. In this equipment we use four tanks for the paints and solvent with the desired pipes, valves and valves to the paint pumping. We also use a high-resolution camera that is linked with Arduino mega micro-controller [11] for the algorithm processing to monitor the mixed paint color. A MATLAB program [12, 13] is used for the algorithm and experimental analysis.

The remaining parts of this work can be summarized as follows. Section II presents the color composition background. This includes how can chromaticity space be useful to our problem formulation. Section III presents the modeling approach with suggested empirical functions that can be used for paint mixing response. A paint mixing algorithm is presented in Section IV. Experimental results are given and discussed in Section V. Finally, the work is concluded in Section VI

2. Color Composition Background

For the purpose of obtaining new paint color with paint mixing process, we have to tackle the coloring theory and its relation to the computer engineering. Several references discuss this issue in detail [14, 15, 16]. One of the very helpful tools in color vision theory is transforming the colors into numerical values, therefore one can easily deal with the type and transparency of the color. Three well-known models are available in the literature to describe the color coding; either composting the color in red, green and blue (RGB) color space, physical model using Kubelka-Munk (KM) theory [17], or defining the any color using red, yellow and blue (RYB) color space. However, computer engineers normally employ the RGB color model in computer graphics which is defined by the three chromaticities of the red, green and blue primaries. The RGB color model is an additive color model as shown in Fig. 1a. Another color model is cyan, magenta, and yellow key model (CMYK) which is a well-known color model for subtractive color composting used in printing [18, 19]. But both RGB and CMYK color models fail to reproduce paint like appearance such as yellow and blue make green. Another RYB color model by Johannes Itten is widely used in art education [20]. In this work we use the RGB system, since one can convert a color system into another color system easily, as demonstrated elsewhere, e.g., to [21]. Fig. 1 shows three color systems, in Fig. 1a shows how one can composite a new color using three color systems; RGB, RYB and CMYK, where Fig. 1b shows the RGB cube.



(a) (b)

Fig. 1. Color spacing system: 1a: Color composition systems; RGB, RYB and CMYK. 1b:bRGB cube.

The RGB cube Fig. 1b illustrates that the red, green and blue values can vary from 0 to 255. For example, to composite the purple color one has to set the red, green and blue to 128, 0 and 128, respectively. These values are called red, green and blue tristimulus values. These values can be easily used with the following vector notation $RGB_{purple} \in \mathrm{R}^3$, namely

$$RGB_{purple} = \left[R_{purple}\ \ G_{purple}\ \ B_{purple}\right]^T = [128\ 0\ 128]^T \quad (1)$$

Another method to represent the RGB coloring system is to use chromaticity space with relative tristimulus values. This is obtained by normalizing the tristimulus values as follows [22]:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}. \quad (2)$$

This results in chromaticity coordinates *r, g* and *b* that are invariant to overall brightness.

By definition $r + g + b = 1$, so one coordinate is redundant and typically only r and g are considered. Since the effect of intensity has been eliminated the 2-dimensional quantity *(r, g)* represents color. Therefore the chromaticity vector for the purple is $rgb_{purple} = [0.5\ 0\ 0.5]^{T}$ or $rg_{purple} = [0.5\ 0]^{T}$.

When it is required to compose the purple color from red, green and blue colors. It is needed to calculate the desired ratios from these colors. Then they are mixed together very well until obtaining the purple. Before mixing the colors, they can be randomly over each other. For example, the green at the bottom, then comes the blue then comes the red, which has the chromaticity vector rgb_{red} = [1 0 0] ^{T}. So that the distance between the red and purple d_{rp} using the color cube Fig. 2, but with relative tristimulus values, can be calculated by following equation:

$$d_{ij} = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2} \quad (3)$$

where the indices i and j refer to the initial and final colors which are in the above example red and purple, respectively. Thus, the distance d_{rp} will be about 0.707.

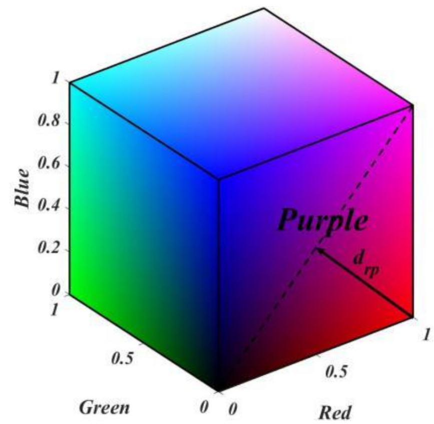


Fig. 2. The distance between the red and purple on the rgb cube with relative tristimulus values.

To investigate this relation, a high-resolution camera is located at the top of the mixing tank to measure the tristimulus values of the surface color. Initially, the distance between the surface color and the color of the final mixed paint is 0.707 in our above example or, alternatively, the distance between the surface color of the unmixed paint and the color that is measured by the camera will be zero before starting to mix. Then this distance increases with mixing until the final mixed paint, with a steady state color will have the distance of 0.707. Furthermore, this distance represents the degree of homogeneity of the paint with respect to the steady state color and, thus, there is no need to continue mixing after this point.

3. Modeling Approach

The dynamic behavior of the mixing processes can be obtained through a plot of dimensionless time-varying distance function (d) from the first instant of mixing. Our next goal is to find the relation of changing in distance with respect to the time. This relation clearly represents the model of mixing. The dynamic behavior of the mixing process can be modeled mathematically (e.g., diffusion and flow fluxes) or empirically based on 'reasonable' exponential models that describe the process of mixing. Empirical models of industrial process engineering can be then used here by employing the observations of experimental data to gain insight into the development of the process [23, 24, 25]. Before assuming the empirical formula that can be well-suited to the paint mixing, it is important to note that, not all the distance observations by the camera will satisfy the following inequality:

$$0 \le d(t) \le d_{ss} \quad (4)$$

where d_{ss} is the final distance between the surface color before mixing and the steady state surface color. This is because that the camera can observe other transient colors with tristimulus values that gives $d > d_{ss}$. Therefore, the formula of transient response must involve the initial distance, which must be zero, final distance which must be d_{ss}, overshoot which will be greater than or equal to d_{ss} and some oscillations until reaching the final steady state distance d_{ss}. Based on this physical understanding of the process Eq. (5) can satisfy Constraint (4) and, at the same times, describes the distance as a time-varying function:

$$d(t) = d_{ss} + e^{-A_1 t}(A_2 \cos(A_4 t) + A_3 \sin(A_4 t)), \quad (5)$$

where A_1 represents the speed of convergence to reach the final desired paint color (i.e. equivalent to some kind of rate coefficient of the process), A_2 and A_3 represent the overshoots with respect to cosine and sine terms, respectively, and A_4 is called the damping speed. Eq. (5) is a solution of a second order differential equation. We can also simplify Eq. (5) by assuming the oscillation happens only with cosine term and shifted with a certain angle, namely:

$$d(t) = d_{ss} + A_5 e^{-A_6 t} \cos(A_7 t + \theta), \quad (6)$$

here the overshoot is only represented by A_5, where A_6 represents the speed of convergence to reach the final desired paint color and A_7 is the damping speed. On the other hand, if there are no oscillations, (i.e., when all the observed distances are below the distance of the mixed paint which satisfy Inequality (4)) then Eq. (6) can be rewritten as follows:

$$d(t) = d_{ss} + A_8 e^{-A_9 t}, \tag{7}$$

where A_8 and A_9 represent the overshoot; which is a kind of distance initial rate, and the speed of convergence to reach the final desired paint color, respectively. Eq. (7) is a simple first order model that is used typically in modeling kinetic chemical processes.

Now, the question whether the observed readings that come from the camera can fit one of the Eqs. (5) -(7) well. This can be answered through the most powerful tool of nonlinear least square method [26]. In the nonlinear least square method, we normally formulate a nonlinear programming problem using the observed data from the camera and one of the time-varying functions in Eqs. (5) -(7). To summarize this procedure, assume we have an observed distance vector from the camera D, namely:

$$D = [d_0\ d_1\ \dots\ d_n]^T \tag{8}$$

where d_k is the distance at k^{th} sampling, $k = 0, 1 \dots, n$. Now to use the nonlinear least square method to fit the distance vector D (cf. Eq. (8)) to the time-varying function, e.g., Eq. (6), a nonlinear programming (NLP) problem will be formulated as follows.

Find the parameters: d_{ss}, A_5, A_6, A_7 and that describe the empirical function $d(t)$, cf. Function (5) and minimize the objective function:

$$L = \sum_{k=0}^{n} (d(t_k) - d_k)^2 \tag{9}$$

here the difference $(t_{k+1} - t_k = \Delta t)$ is the sampling time of the camera. This objective function can be also used for empirical Functions (5) and (7), by finding the optimal describing parameters therein. The above NLP problem can be solved easily using NLP solver. We use the well-known fitting MATLAB subroutine lsqcurvefit [27] to solve this NLP problem.

4. Paint Mixing Algorithm

In any mixing processes, the suitable ratios of red, green and blue must be mixed together to produce the desired color. Thus, a mixing algorithm to calculate the suitable ratios is demonstrated here. Before calculating color ratios, the quantity of the final product must be also given by the user. For simplicity, we will use three paints with similar viscosity and density. That is the quantity can be given by the mass or volume since the mass of the paint *(m_h)*, where the index h refers to the tristimulus values of the red, green or blue colors, is given by

$$m_h = \rho_h V_h \tag{10}$$

where h is the paint density and V_h is the paint volume. Therefore, the total volume of each paint h can be calculated by

$$V_h = x_h V_T \tag{11}$$

where V_T is the total volume that is given by the user and is the volume fraction of the color h. This also results in volume additivity, i.e., $V_T = V_r + V_g + V_b$.

In accordance we use Algorithm 1 to implement and observe the data measurements for the paint mixing process:

Algorithm 1: Paint mixing algorithm

Inputs:	The total required paint volume, the required color, i.e., the tristimulus values R, G and B, homogeneity tolerance tol.
Outputs:	The volumes of red, green and blue paints, the required mixed paints, the parameters d_{ss}, A_i , $i=1,\dots,9$, θ (cf. Eqs. (5)-(7)), mixing time that guarantees a color homogeneity that is with certain accepted tolerance tol.
Step 0:	**Initialize**: The input paint colors $rgb_r = [1\ 0\ 0]^T,\ rgb_g[0\ 1\ 0]^T,\ rgb_b = [0\ 0\ 1]^T$
Step 1:	Compute the relative tristimulus values of the desired paint color; $rgb_h = [r_h\, g_h\, b_h]^T$using Eq. (2)
Step 2:	Compute volume of the input paints using Eq. (11), i.e, $V_L = V_r + V_g + V_b$.
Step 3:	Pump the computed volumes in **Step 2** into the mixing tank, simultaneously.
Step 4:	• Set the counter $k = 0$. Initialize the distance $d_k = 0$. • Observe the tristimulus values of the surface color, i.e. $rgb_0 = [r_0\ g_0\ b_0]^T$, using the top camera. • Start mixing.
Step 5:	• Set $k = k + 1$. Observe the tristimulus values $rgb_k = [r_k\ \ g_k\ b_k]^T$ of the surface color. • Compute the distance between rgb_0 and rgb_k, i.e. $d_k = rgb_k - rgb_0$ using Eq. (3). • Set $d_k = [d_{k-1}\ \ d_k]$
Step 6:	• If $\lvert rgb_h - rgb_k \rvert > $ tol; go to **Step 5** • Else, stop mixing.
Step 7:	Use the subroutine lsqcurvefit to fit the vector d_k with Functions (5), (6) or (7).
Step 8:	**Out**: Mixing time= $k\Delta t$, the parameters $d_{ss}, A_i, i = 1, \dots 9, \theta$. **Exit**.

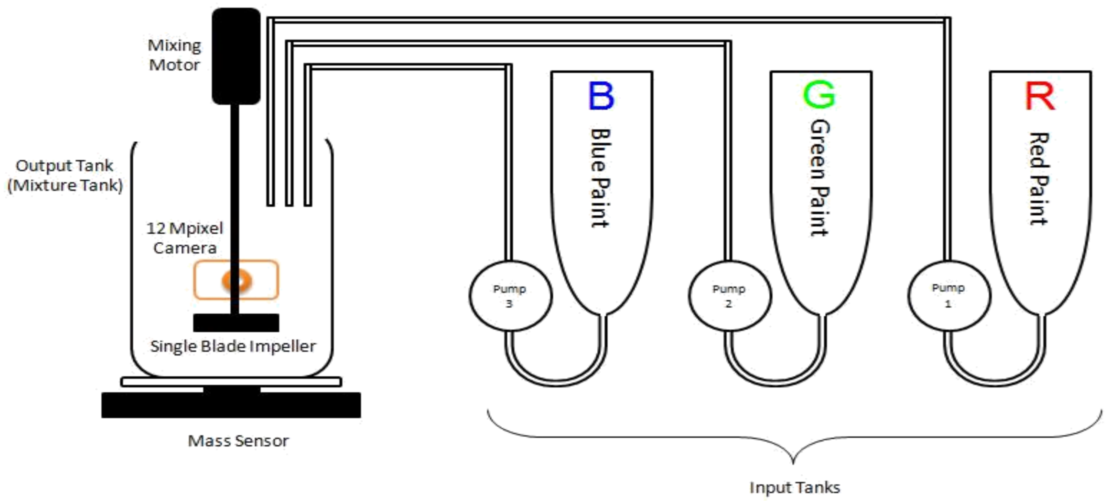
5. Experimental Results

The proposed work is implemented and tested using a lab-made equipment. In this equipment we use four transparent tanks; one for mixing and three for the raw paints, with dedicated pumps, pipes and valves as shown in Fig. 3. A single blade impeller, which is derived by 12 V DC motor, is used for mixing. The mixing speed varies from 20 to 200 rpm. A 12 M pixel camera, an Arduino Mega micro-controller and an I7 laptop machine are used for paint imaging, algorithm implementation and data regression (fitting), respectively. To guarantee that the pumped paints have an identical density, we also use a mass sensor with a desired control loop beyond the mixing tank attributes.

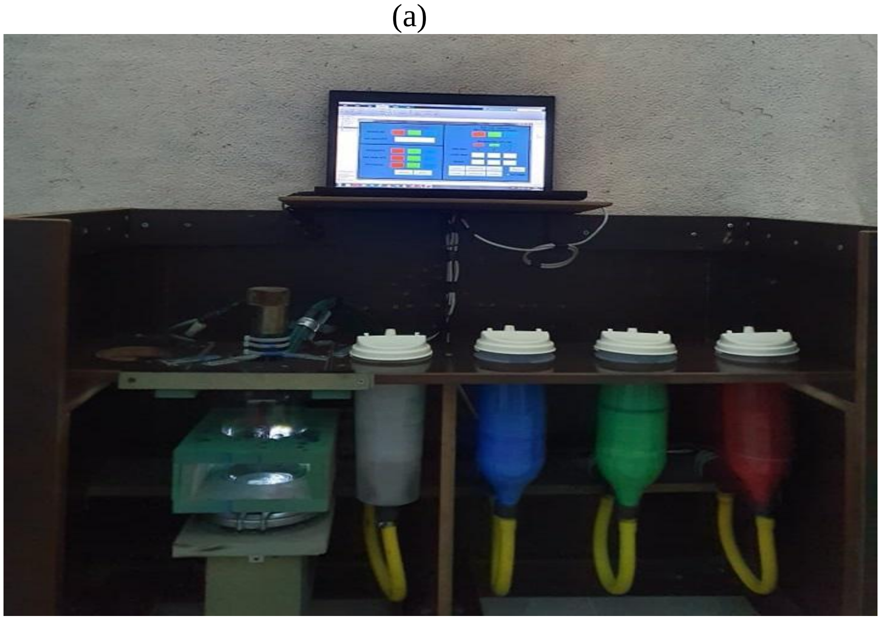
For the raw paints, we used Spies SI-Tone paints [28]. This paint is a water-based paint with 1.18 g/mm3 density; i.e., it can be formulated, diluted and cleaned up with water.

To guarantee that all experiments will give reliable results, they were done at same pre-specified and 'suitable' luminous intensity, so that the tristimulus values have the same accuracy when capturing the paint images.

Several experiments were done to test our proposal. Each experiment aims to produce a 300 mL of a new desired paint. The first set of experiments is intended to produce x_1 color with tristimulus vector $B_{x_1} = [180\ 210\ 100]^T$, at maximum mixing motor speed of 200 RPM. To ensure the validity of color mixing Algorithm 1 and to choose the proper fitting function we repeat this experiment four times, that is the four experiments are done to produce the same color with a same mixing speed. Fig. 4 shows the result of these experiment. Here we see the measured distance as function with the time (black) where the fitting curves cf. Eqs. (5), (6) and (7) are in blue, green and red, respectively.



(a)



(b)

Fig. 3. Paints mixing equipment: (3a): Equipment setup, (3b): Lab-made equipment.

Table I
Comparison between the responses of mixing paints with $RGB_{x_1} = [180\ \ 210\ \ 100]^T$

Experiment		1			2	
Empirical function	Eq. (7)	Eq. (6)	Eq. (5)	Eq. (7)	Eq. (6)	Eq. (5)
Objective value (L) cf. Eq. (9)	0.1731	0.1257	0.0741	0.1731	0.0904	0.0462

Experiment		3			4	
Empirical function	Eq. (7)	Eq. (6)	Eq. (5)	Eq. (7)	Eq. (6)	Eq. (5)
Objective value (L) cf. Eq. (9)	0.1619	0.1361	0.1007	0.2069	0.1697	0.1205

From Table 1, we see that the best empirical function in this case is Function (5), since it has the minimum objective function[1] less than others.

The second set of experiments was performed to investigate the effect of the mixing speed to the distance response 'homogeneity response'. In this set, it also is intend to produce x_1 color, i.e., with tristimulus vector $RGB_{x_1} = [180\ 210\ 100]^T$ but, however, with different mixing speeds, namely, 200, 160, 120 and 80 rpm.

Fig. 5 shows the time-varying distance response according the measured distance (black) and empirical Function (5) with the previous speeds, respectively. Here we can see, a slower mixing speed results more delay to reach the homogenous paint. Obviously, increasing the mixing speed increases the rate of mass transfer between regions of various color. Thus, this decreases the time required to reach color homogeneity. This is well reported in chemical literature.

The third set of experiments was done to compare the mixing responses for producing different colors with the same mixing speed. Fig. 6 shows the responses of mixing to produce four colors with tristimulus vectors $RGB_{x_1} = [180\ 210\ 100]^T$, $RGB_{x_1} = [90\ 100\ 215]^T$ and $RGB_{x_1} = [180\ 100\ 100]^T$, respectively.

---

[1]We use the default settings and properties of the 'lsqcurvefit' subroutine with disregard to NLP initial guess [27].

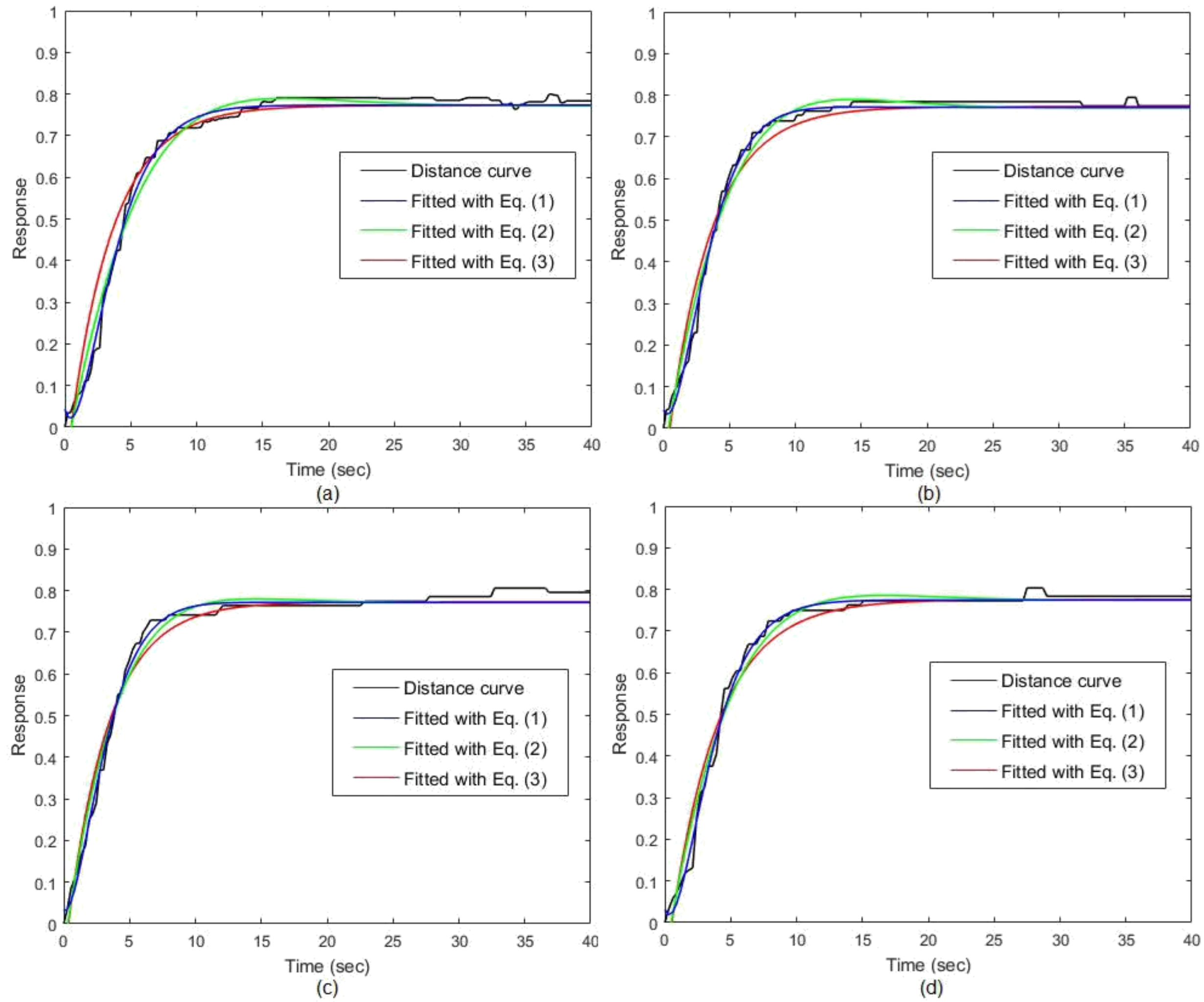


Fig. 4. The result of producing the color with the $RGB_{x_1} = [180 \;\; 210 \;\; 100]^T$ (repeated four times: a, b, c and d). Black: measured distance curve. Blue, green and red curves: response with fitting Functions (5)-(7), respectively.

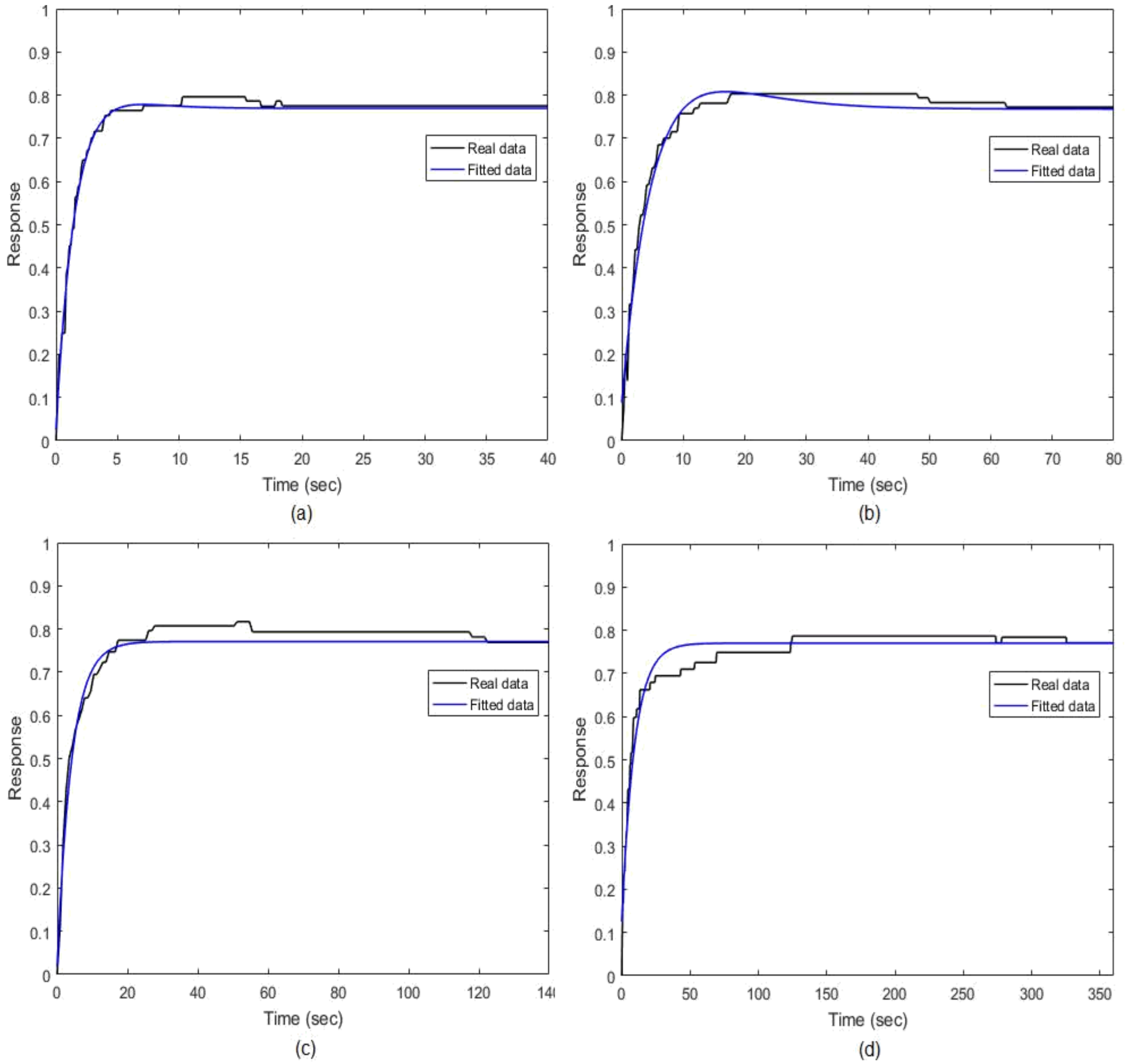


Fig. 5. The result of producing the color with the $RGB_{x1} = [180\ 210\ 100]^T$ with different mixing speeds 200, 160, 120 and 80 rpm: a, b, c and d, respectively. Black: measured distance curve. Blue: response with fitting Function (5)

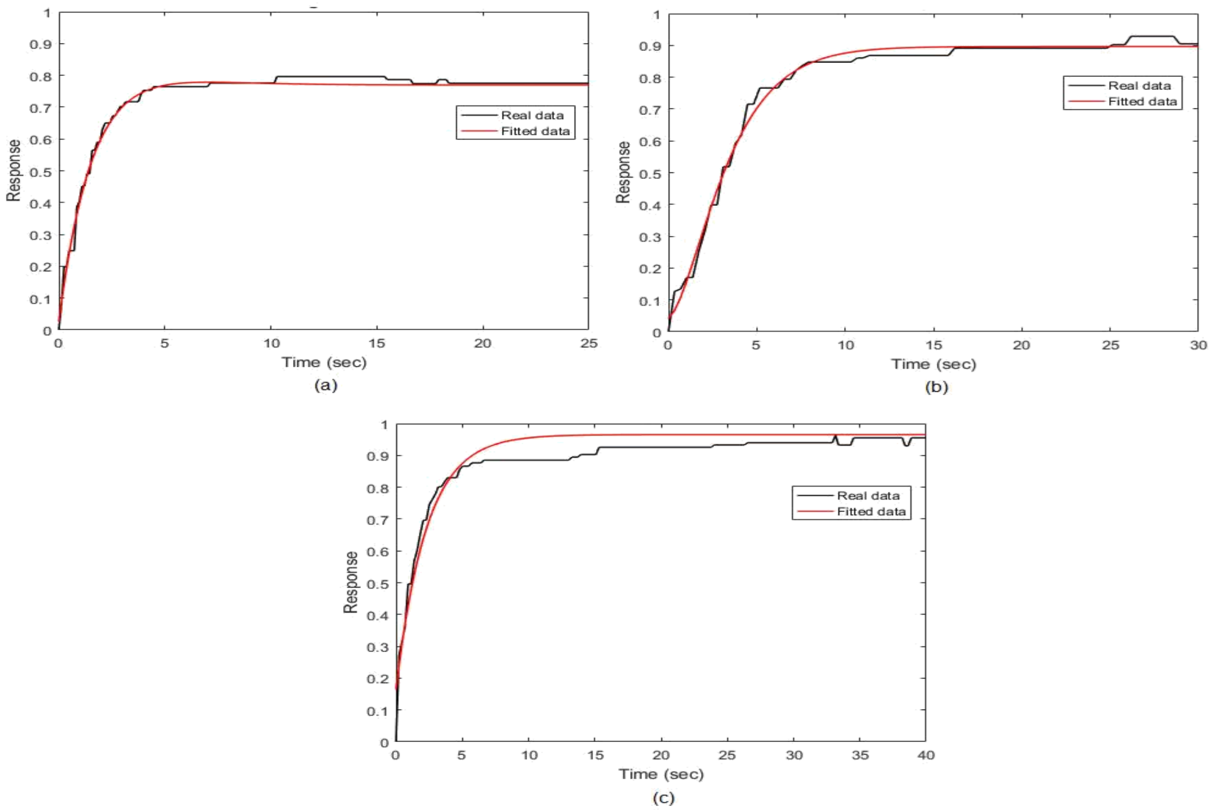


Fig. 6. The result of producing the colors with the $RGB_{x_1} = [180\ \ 210\ \ 100]^T$, $RGB_{x_1} = [90\ \ 100\ \ 215]^T$ and $RGB_{x_1} = [180\ \ 100\ \ 100]^T$, in a, b and c, respectively, with mixing speed 200 rpm. Black: measured distance curve. Blue: response with fitting Function (5)

Table II shows the parameters of experiments Sets 1, 2 and 3. As shown in Section 3. In these experiments, we are interested in the steady state distance with speed of convergence to reach the homogenous paint as well as the overshoot terms which are represented through the parameters d_{ss}, A_1, A_2 and A_3. Here also we can see that the mixing motor speed can affect the speed of convergence to the steady homogenous paint color, but, however, when different paints are intended to produce different colors, as in experiments Set 3, minor effect on the speed of the convergence can be noticed. The reason of this can be attributed to the distance between the surface color and the desired color is not a physical distance, but it is a distance due to tristimulus vector in chromaticity space.

Table II
Parameters of Empirical Function (5) for the experiments Sets 1-3

Figure	Experiment	Mixing time (second)	Distance	d_{ss}	A_1	A_2	A_3
4a	1	21.56	0.7737	0.7732	0.502	-0.73	-4.7458
4b	2	23.19	0.7737	0.7709	0.491	-0.7265	-2.0178
4c	3	22.53	0.7737	0.7716	0.516	-0.7411	1.9839
4d	4	21.15	0.7737	0.7745	0.502	-0.7427	-3.4092
5a	1	21.17	0.7737	0.7696	0.498	-0.7437	2.9998
5b	2	76.29	0.7737	0.768	0.1455	-0.6797	7.7353
5c	3	40.82	0.7737	0.7709	0.244	-0.7516	0.3581
5d	4	91.42	0.7737	0.7701	0.106	-0.6446	-0.3525
6a	1	21.17	0.7737	0.7696	0.499	-0.7437	2.9998
6b	2	22.04	0.8931	0.8964	0.538	-0.855	-5.9717
6c	3	21.04	0.9746	0.9654	0.434	-0.8004	0.1999

6. Conclusion

In this work, a paint mixing algorithm for water paints was presented. It was presented using pseudocode. Using this algorithm, we can successfully determine the wanted paint color and the volume or mass of the input paints. Then our paint mixing machine that was designed for this purpose will use this algorithm to compute the required raw paints and then mix them for an enough period to produce a fully homogamous paint color.

In the paint mixing machine, A Lab-mad equipment was built using four paint tanks with the desired pumps, pipes, micro-controller and high-resolution camera. One of the paint tanks was used for paint mixing using single blade impeller that is coupled with a speed-controlled DC motor.

After paint mixing, we can successfully use one of the empirical functions to describe the color homogeneity response that is occurred during the mixing process. To do this step we have solved the nonlinear programming problem that described by the nonlinear least square problem which is formulated using the empirical function and observed measurement from the camera.

Three sets of experiments were done using this lab-made paint mixing machine to show the different responses of the paint mixing as well as the effectiveness proposed algorithm. The first set was applied to choose the best empirical function among three suggested functions. The second set was done to show the effect of the mixing speed to the paint homogeneity. While the third set is done to show the mixing behaviors when mixing different paints to get different colors. Furthermore, also this algorithm was applied using RGB chromaticity space, but, however, it can be problems free applied for other chromaticity space like RYB and CMYK.

This approach for investigating the hydrodynamics of color mixing process can be generalized for investigating the kinetic of various physicochemical processes in liquid phases. Such an investigation is the subject of our future work.

References

[1] B. Roffel and B. Betlem, Process dynamics and control: modeling for control and pre-diction. John Wiley & Sons, 2007.

[2] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, and F. J. Doyle III, Process dynamics and control. John Wiley & Sons, 4 th ed., 2017.

[3] L. Schiop and M. Giceanu, Mathematical modelling of color mixing process and PLC control implementation by using human machine interface," 2010 3rd International Symposium on Electrical and Electronics Engineering (ISEEE), pp. 165-170, 2010.

[4] G. Towler and R. Sinnott, Chemical engineering design: principles, practice and eco-nomics of plant and process design. Elsevier, 2012.

[5] S. Brindha, P. Kishorniya, R. Manickam, K. N. Chakkaravarthy, and C. Poomani, Automated color mixing machine using arduino," International Journal of Engineering Research & Technology, vol. 6, pp. 1-4, 2018.

[6]Hassan, Color mixing machine using plc and scada," WSEAS Transactions on Systems and Control, vol. 10, pp. 650-665, 2015.

[7] S. Sanamdikar and C. Vartak, Color making and mixing process using PLC," International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 2, no. 5, pp. 170-174, 2013.

[8] M. A. Muftah, A. M. Albagul, and A. M. Faraj, Automatic paint mixing process using LabVIEW," Mathematics and Computers in Science and Industry, pp. 233-238, 2014.

[9] M. Al-Jabari, M. Abualfailat, and S. Shaheen. Treating leather tanning wastewater with stone cutting solid waste," Clean-Soil, Air, Water, vol. 40, no. 2, pp. 206-210, 2012.

[10] M. Jabari, F. Aqra, S. Shahin, and A. Khatib, Monitoring chromium content in tannery wastewater," The Journal of the Argentine Chemical Society, vol. 97, no. 2, pp. 77-87, 2009.

[11] M. Micro-controller, Retrieved 01-01-2020. https://www.arduino.cc/.

[12] K. Al-Malah, MATLAB Numerical Methods with Chemical Engineering Applications. McGraw-Hill Professional, 2013.

[13] MATLAB Solutions, Retrieved 01-01-2020. https://www.mathworks.com.

[14] R. Cipolla, S. Battiato, and G. M. Farinella, Computer Vision: Detection, recognition and reconstruction, vol. 285. Springer, 2010.

[15] T. De Bie, N. Cristianini, and R. Rosipal, Handbook of geometric computing: Applications in pattern recognition, computer vision, neural computing, and robotics, chapter eigenproblems in pattern recognition," 2005.

[16] T. Gevers, A. Gijsenij, J. Van de Weijer, and J.-M. Geusebroek, Color in computer vision: fundamentals and applications, vol. 23. John Wiley & Sons, 2012.

[17] W. E. Vargas and G. A. Niklasson, Applicability conditions of the Kubelka-Munk theory," Applied optics, vol. 36, no. 22, pp. 5580-5586, 1997.

[18] D. McGavin, B. Stukenborg, and M. Witkowski, Color gures in BJ: RGB versus CMYK," Biophysical Journal, vol. 88, no. 2, pp. 761-762, 2005.

[19] P. Nishad, Various colour spaces and colour space conversion," Journal of Global Re-search in Computer Science, vol. 4, no. 1, pp. 44-48, 2013.

[20] J. Itten and E. van Haagen, The Art of color: the subjective experience and objective rationale of color. Van Nostrand Reinhold, 1991.

[21] J. Sugita and T. Takahashi, RYB color compositing," Proc. IWAIT, poster, 2015.

[22] P. Corke, Robotics, vision and control: fundamental algorithms in MATLAB R second, completely revised, vol. 118. 2nd ed., 2017.

[23] C. Damgaard, Evolutionary ecology of plant-plant interactions: an empirical modelling approach. ISD LLC, 2005

[24] S. Lakshminarayanan and H. Takada, Empirical modelling and control of processes with recycle: some insights via case studies," Chemical Engineering Science, vol. 56, no. 11, pp. 3327-3340, 2001.

[25] M. Shin and A. L. Goel, Empirical data modeling in software engineering using radial basis functions," IEEE Transactions on Software Engineering, vol. 26, no. 6, pp. 567- 576, 2000.

[26] Griva, S. G. Nash, and A. Sofer, Linear and nonlinear optimization, vol. 108. Siam, 2009.

[27] M. O. T. Release, The MathWorks, Inc. "lsqcurvefit solver" Retrieved 10.10. 2020.https://www.mathworks.com/.

[28] S. SI-Tone, Retrieved 07-01-2020. http://www.sipesworld.com/en

Majdi Salman received the B.Eng. degree in mechatronics engineering from Palestine Technical University (PTU) and he is now a master student at Palestine Polytechnic University.

His research interest s are mechatronics systems and control systems.

Jasem Tamimi received the B.Eng. degree in electrical engineering in 2002, M.Sc. degree in electronic and computer engineering in 2006, and PhD degree in computer engineering in 2011 from Palestine Polytechnic University, Alquds University and Ilmenau of Technical University, respectively. From 2011 until now he is with department of mechanical engineering at Palestine Polytechnic University. His research interests are model predictive control, optimal and nonlinear control.

Maher Al-Jabari is a full professor of chemical and environmental engineering in the department of mechanical engineering at Palestine Polytechnic University. Al-Jabari has an experimental research record in separation processes. He also developed various mathematical models for transport processes in chemical and environmental applications. He has a research record in the field of waterproofing and durability of cementitious construction materials. He received his PhD in 1994 from McGill University, Montreal, Canada.