

Seamless Manual-to-Autopilot Transition for Robotic Welding*

Haruhiko Eto¹ and H. Harry Asada, *Member, IEEE*¹

Abstract—We propose a novel approach to human-robot shared control for arc welding. In current robotic welding, a human operator has to input a lot of parameters, including feed rate, swing width, and frequency, by using a teaching pendant or a control panel before executing the task. For manufacturing workers this on-site teaching is tedious, time-consuming, and unintuitive. In this paper, a method is developed for acquiring those parameters directly from a human demonstration and seamlessly transitioning from manual operation to automatic control. With this method, a welding worker directly executes a welding task, and the motion of a welding torch is observed, from which key parameters are identified and the machine performs the rest of the task autonomously. No tedious parameter input is required, but the worker can jump-start the task. The motion of a welding torch is represented as a combination of sinusoidal and linear functions. Discrete Fourier Transform (DFT) and Recursive Least Squares (RLS) estimates are used for identifying the parametric model in real-time. Furthermore, an algorithm is developed for determining whether sufficient data have been obtained and when to switch from manual operation to autonomous control. The method is implemented on a virtual teleoperation system and seamless control transition is demonstrated.

I. INTRODUCTION

Welding is one of the most critical manufacturing processes in heavy industries. Shipbuilding, for example, requires a number of welding operations. Fully autonomous welding machines are still impractical and difficult to use due to the complexity of the environment and variety of workpieces. Portable semi-automatic welding carriages have been used as a practical solution for fillet welding and butt welding instead of expensive intelligent robots [1]. These carriages can track a welding joint at a constant speed with drive wheels and a mechanical guiding system (for instance, self-guiding rollers or guide rails). Thus, these carriages can perform continuous welding along a straight or gentle curved joint. Commercialized welding carriages usually have 1 or 2 active DoF: drive wheels, for moving along the welding joint line, and a torch swing mechanism, for making a specific rotating/swinging motion so-called weaving motion. Robotic manipulators, which typically have more than 5 DoF, are another option for this work situation. These manipulators are installed on mobile bases or other mechanisms such as gantries and over-head cranes, and they have sensors for welding seam tracking [2].

In both the case of welding carriages and the case of manipulators, a human welder practically has to teach those

devices initial welding parameters to perform the welding task autonomously. Our target applications are Metal Inert Gas (MIG) welding and Metal Active Gas (MAG) welding, in which shielding gas and an electrode wire are automatically fed to the welding torch from external equipment. The gas flow rate, wire feeding speed, and arc current are usually pre-programmed by the welder. The welding torch motion parameters such as traveling speed, swing width, and frequency require more flexibility, because these values vary depending on workpiece geometry and have to be adjusted by the welder. These motion parameters are usually set by manually turning dials and pushing buttons on the control panel of the devices or other type of teaching tool. This parameter setting process provides less flexibility for different workpieces and is not intuitive for human welders; therefore, they often can input wrong values and then have to modify them.

Autonomous welding has been studied for a long time. Early works by D. Hardt and his collaborators aimed to regulate a welding process with closed-loop control and adaptive and learning methods [3]–[7]. Complete autonomy, however, is difficult due to various factors as addressed above. More recent approaches include a human in the loop. Teleoperated welding systems have been developed by several authors [8]–[10], and shared control approaches have been investigated [11]. Furthermore, human welding skills have been measured and analyzed offline [12], and used for autonomous control.

This paper proposes a different type of shared control, where manual welding operation and autonomous control are seamlessly switched in real-time. This eliminates tedious on-site parameter setting, and allows a robotic welder to execute the task as intended by the human welder. Once autonomous mode activates, the human welder is released and can work on other tasks. In the following sections, the seamless manual-to-autopilot transition is detailed, and algorithms are developed for analyzing human data and switching the control mode without stopping the process.

II. APPROACH: SEAMLESS MANUAL TO AUTOPILOT TRANSITION

We propose a shared control where manual and autonomous control are time-shared, focusing on a seamless transition from manual to autopilot mode. In this approach, a skilled human welder manually performs a welding operation for the first few segments. A real-time estimation algorithm will be developed to analyze the data and identify the intended operation of the human welder. As soon as all the parameters have been identified, the system takes over the

*This work was supported by Sumitomo Heavy Industries, Ltd.

¹The authors are with the the d'Arbeloff Laboratory for Information Systems and Technology in the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA
eto@mit.edu

welding operation and completes the task autonomously. This seamless transition from manual to autonomous operations has a few distinct features:

- A human welder can jump-start a welding operation without inputting any parameters. Optimal welding parameters are not explicitly known for human welders, but they can demonstrate best operations by manipulating the torch. This can make the human-robot interface intuitive and easy to use.
- As addressed previously, the parameter setting method of today's welding carriages and welding robots is not flexible. The proposed method can realize a broader class of torch motions and patterns, so that a human welder can express subtle movements and skillful operations.
- The human demonstration for teaching and the autonomous task execution are seamlessly connected. Once the welding starts, it never stops.
- The human welder is released from the welding operation, once the robot confirms that it has gained enough information from the human. The human can work on other welding robots to increase productivity.

Fig. 1(a) shows one of the realizations of the above method. The human welder holds a real welding torch and performs a welding operation. The welding torch is also attached to a robot, which is back-drivable by design, or controlled in zero-G mode, so that the human can freely move the torch. The movement of the torch is measured with the sensors on the robot (e.g. encoders of the joints). From the measured data, its trajectory and motion pattern are identified in real-time. As a sufficient amount of data is obtained, the robot takes over the task to complete the welding. Fig. 1(b) shows another realization where the human welder uses a mock-up welding torch that is instrumented with position sensors so that the motion of the torch is measured. The measured human movement is transmitted to the welding robot that reproduces the human motion. As before, the robot takes over the task once it gains enough data. This realization is a type of teleoperation, requiring a virtual environment so that the human welder can observe the weld pool and the workpieces being joined. In both cases, the robot must be able to track the welding line autonomously with sensors that detect the welding line.

III. METHOD

In this section, we present a parametric model of welding torch motion and develop an algorithm for estimating the parameters from a human welder's movements.

A. Welding Trajectory Model

The position and orientation of the welding torch is represented with a vector defined as

$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T = [x \ y \ z \ \alpha \ \beta]^T$$

See Fig. 2. We assume that the welding line is piecewise straight and uniform, so that the required torch motion does not change in each segment of straight line. Typically a human welder moves the torch at a constant traveling speed,

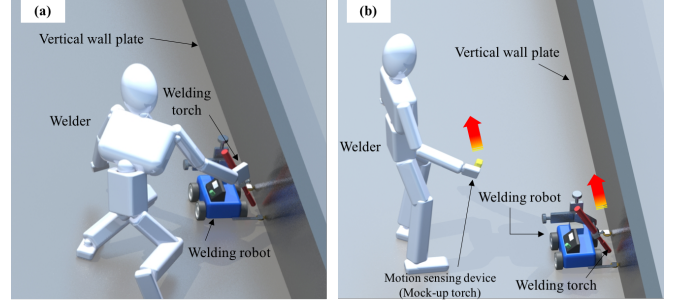


Fig. 1. Two examples of the use of our proposed method. In this situation, a human welder and a welding carriage are performing a continuous fillet welding. The carriage moves along the welding joint by using its roller guiding mechanism, which is contacting to the vertical plate. In (a), the welder directly moves the torch attached to the robot. The movement of the torch is measured and its motion pattern are identified in real-time for automatic welding. In (b), the robot is teleoperated by the welder's hand motion by using a motion sensing device.

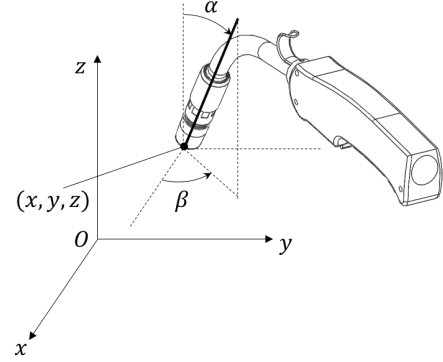


Fig. 2. The position and orientation of the mockup torch.

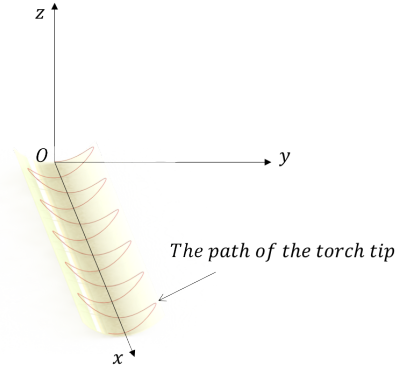


Fig. 3. An example of three-dimensional weaving path patterns. The travel direction is x axis and the swing direction is along y axis. The torch is swinging in z direction as well.

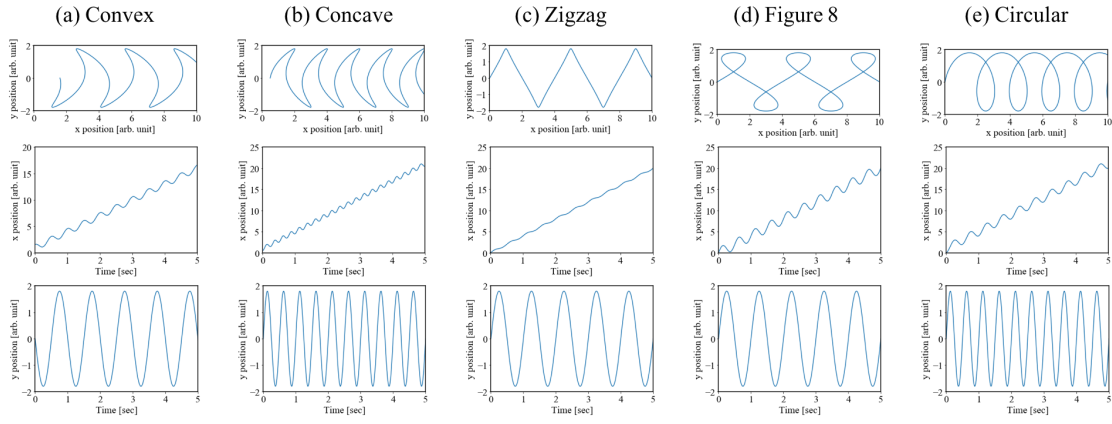


Fig. 4. Examples of weaving path patterns projected onto a two-dimensional plane, which can be represented by our proposed model, and their trajectories in x and y directions as a function of time t . The x axis is the traveling direction and the y axis is the swing direction. (a) shows a convex path and its trajectories, $x(t) = 0.6 \sin(2\pi \cdot 2.0 \cdot t + \pi/2) + 3.0t + 1.0$ and $y(t) = 1.8 \sin(2\pi \cdot 1.0 \cdot t + \pi)$. (b) shows a concave path and its trajectories, $x(t) = 0.5 \sin(2\pi \cdot 4.0 \cdot t + 3\pi/2) + 4.0t + 1.0$ and $y(t) = 1.8 \sin(2\pi \cdot 2.0 \cdot t)$. (c) shows a zigzag path and its trajectories, $x(t) = 0.2 \sin(2\pi \cdot 2.0 \cdot t - 4.0\pi/2) + 4.0t$ and $y(t) = 1.8 \sin(2\pi \cdot 1.0 \cdot t)$. (d) shows a figure 8 path and its trajectories, $x(t) = 1.2 \sin(2\pi \cdot 2.0 \cdot t - 4.0\pi/2) + 4.0t$ and $y(t) = 1.8 \sin(2\pi \cdot 1.0 \cdot t)$. (e) shows a circular path and its trajectories, $x(t) = 1.0 \sin(2\pi \cdot 2.0 \cdot t - \pi/2) + 4.0t + 1.0$ and $y(t) = 1.8 \sin(2\pi \cdot 2.0 \cdot t)$.

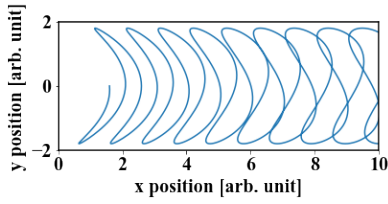


Fig. 5. A convex path curve moving along the x direction with the frequency ratio $f_x/f_y = 2.02$. The small difference from the standard ratio collapses the curve shape.

or feed rate, combined with an oscillatory motion: weaving. We thus model each torch's position and orientation $x_i(t)$ as a combination of a sinusoidal function and a linear function:

$$x_i(t) = A_i \sin(2\pi f_i t + \psi_i) + a_i t + b_i \quad (1)$$

where A_i is amplitude, f_i is frequency, ψ_i is phase angle, a_i is linear velocity, and b_i is a constant bias. We assume that the x axis and y axis correspond to the welding travel direction and the swing direction, respectively.

This model $x(t)$ can represent stringer bead path and typical weave bead paths, convex, concave, figure 8, circular, and so on. Fig. 3 shows examples of three-dimensional torch paths, which can be represented by (1). Fig 4 shows examples of weaving patterns projected onto the xy -plane. The time profiles of $x(t)$ and $y(t)$ forming those patterns are also shown in the figure. To form these path curves, the following two conditions must be satisfied. First, $y(t)$ must be symmetric with respect to its center line, which means $a_2 = 0$. Second, the ratio of frequencies f_x/f_y must exactly be $f_x/f_y = 2$ for patterns (a), (b), (c), and (d), and $f_x/f_y = 1$ for pattern (e). Pure sinusoidal curves, which are not involved in Fig. 4, are also an example of typical welding paths and its frequency ratio is $f_x/f_y = 0$. If the frequency ratio is slightly different from these standard values, the path shape will gradually collapse with time in Fig. 5, which

shows a convex path with the frequency ratio $f_x/f_y = 2.02$. This matter will be considered in Subsection C.

The trajectories (1) can be represented in a discrete-time form with a constant sampling interval Δt and a sample index k as follows:

$$x_i(k) = A_i \sin(2\pi f_i \Delta t k + \psi_i) + a_i \Delta t k + b_i \quad (2)$$

This trajectory function can be re-parameterized as:

$$\begin{aligned} x_i(k) &= A_i \sin(2\pi f_i \Delta t k + \psi_i) + a_i \Delta t k + b_i \\ &= \begin{bmatrix} s_{i,k-1} & c_{i,k-1} & k-1 & 1 \end{bmatrix} \begin{bmatrix} \theta_{i,1} \\ \theta_{i,2} \\ \theta_{i,3} \\ \theta_{i,4} \end{bmatrix} \\ &= \varphi_i^T(k) \theta_i \end{aligned} \quad (3)$$

where

$$s_{i,k-1} = \sin(2\pi f_i \Delta t (k-1)) \quad (4)$$

$$c_{i,k-1} = \cos(2\pi f_i \Delta t (k-1)) \quad (5)$$

$$\theta_{i,1} = A_i [\cos(2\pi f_i \Delta t) \cos \psi_i - \sin(2\pi f_i \Delta t)] \quad (6)$$

$$\theta_{i,2} = A_i [\sin(2\pi f_i \Delta t) \cos \psi_i - \cos(2\pi f_i \Delta t)] \quad (7)$$

$$\theta_{i,3} = a_i \Delta t \quad (8)$$

$$\theta_{i,4} = a_i \Delta t + b_i. \quad (9)$$

$x_i(k)$ corresponds to the torch tracking data stream obtained by a human demonstration. The problem is to estimate unknown parameters in (3), namely the frequency f_i and the parameter vector θ_i . Considering its nonlinearity, our algorithm estimates f_i using DFT prior to estimating θ_i with RLS. Once the value of f_i is estimated, the RLS estimation algorithm can be executed.

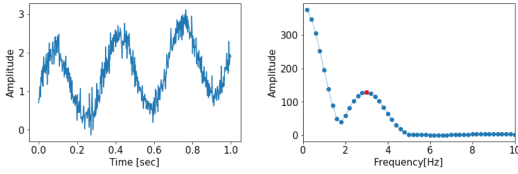


Fig. 6. The left graph shows a synthetic test signal, which consists of a sinusoid and a linear component with random noise. The frequency is 3 Hz and the data length is 1 second. The right graph shows the frequency estimation result after zero padding. The result has two peaks in the graph. The peak around 0 Hz is because of the linear components, and the other peak, marked with a red dot, is the answer value.

B. DFT and Frequency Correction

Since DFT needs a certain number of data points, the tracking data have to be stored in a buffer. The data length can be a few seconds when the frequency of swing motion is, for example, around 1 Hz. The Hann window is applied to the stored dataset and zero padding is used to increase the resolution of estimated frequency. Because the tracking data have a linear component, the DFT result has two peaks as shown in Fig. 6. The peak around 0 Hz is because of the linear components, and then the other peak is the answer. The resultant peak value is identified as the estimated frequency. As discussed in the previous subsection, the frequency ratio f_x/f_y must exactly be 0, 1, or 2. After f_x and f_y are estimated, their ratio is computed and classified to the closest standard value by thresholding. f_x is then replaced as $f_x = 0$, $f_x = f_y$, or $f_x = 2f_y$. The estimated result of f_y by DFT is usually more accurate than that of f_x since $y(t)$ has no linear components; this is why we replace f_x with a value defined by f_y . Thus we finally obtain estimated f_x and f_y .

C. RLS-based Parameter Estimation

Once the estimated frequencies are obtained, the unknown parameter vector θ_i can be estimated by the RLS. Prior to executing the RLS, initial estimates of the parameters are obtained from the data used for the DFT computation. The batch least squares algorithm is applied to the data, in which the frequency is replaced by the estimated frequency \hat{f}_i obtained by DFT. This allows us to provide initial values for the unknown parameters, i.e. $\hat{\theta}_i(0)$, and expedite the convergence of RLS computation. We apply the following RLS algorithm with forgetting factor to estimate $\hat{\theta}_i(k)$:

$$\begin{aligned} \hat{\theta}_i(k) &= \hat{\theta}_i(k-1) \\ &+ \frac{\mathbf{P}_i(k-1)\boldsymbol{\varphi}_i(k)}{\rho_k + \boldsymbol{\varphi}_i^T(k)\mathbf{P}_i(k-1)\boldsymbol{\varphi}_i(k)} [x_i(k) - \boldsymbol{\varphi}_i^T(k)\hat{\theta}_i(k-1)] \end{aligned} \quad (10)$$

$$\mathbf{P}_i(k) = \frac{1}{\rho_k} \left[\mathbf{P}_i(k-1) - \frac{\mathbf{P}_i(k-1)\boldsymbol{\varphi}_i(k)\boldsymbol{\varphi}_i^T(k)\mathbf{P}_i(k-1)}{\rho_k + \boldsymbol{\varphi}_i^T(k)\mathbf{P}_i(k-1)\boldsymbol{\varphi}_i(k)} \right] \quad (11)$$

where $\mathbf{P}_i(k)$ is a covariance matrix, ρ_k is a forgetting factor ($0 < \rho_k \leq 1$), and $\mathbf{P}_i(0)$ is a positive definite matrix. Once we obtain $\hat{\theta}_i(k)$, the reference trajectory can be generated

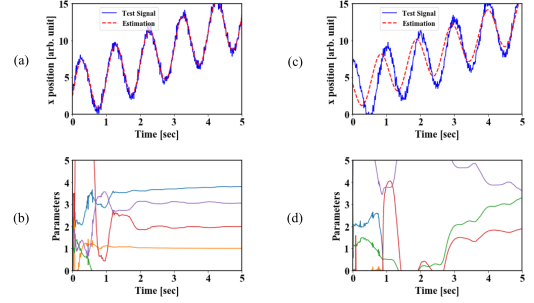


Fig. 7. Parameter estimation results by the EKF. The EKF algorithm is applied to a function consisted of a sinusoid and a linear line with a random noise. (a) $x(t) = 4 \sin(2\pi \cdot 1.0 \cdot t) + 2.0t + 3.0 + \text{noise}$ (b) $x(t) = 4 \sin(2\pi \cdot 1.0 \cdot t + 5/2 \cdot \pi) + 2.0t + 3.0 + \text{noise}$.

as:

$$\hat{x}_i(k | \hat{\theta}_i) = \boldsymbol{\varphi}_i^T(k) \hat{\theta}_i. \quad (12)$$

The reference trajectory \hat{x}_i is fed to the controller of the robotic welding system so that the welding torch follows it. Because of the forgetting factor, the RLS algorithm outputs the parameters that gives more weight to recent data and outputs better fitting results than the old data. This makes a smooth connection between the input trajectory and the reference trajectory. In order to validate the reference trajectory reproduced by the RLS estimation, we observe both the changes in all the estimated parameters and the squared errors of the paths, i.e. $(\hat{x}_i - x_i)^2$. It is considered that the estimation is completed when these values become lower than a cutoff threshold. Evaluating the squared errors is important to a smooth transition from the input trajectory to the reference trajectory, but it does not guarantee the convergence of the estimation. That is why we also monitor the estimated parameters.

There are several online algorithms for nonlinear parameter estimation, such as Extended Kalman Filter (EKF). As shown in Fig. 7(a) and (b), The EKF algorithm is able to compute the unknown parameters of a function, which consists of a sinusoid and a linear function. The EKF requires no batch processing, unlike the DFT combined with the least squares estimation, so the estimation can be completed quickly, in a few seconds. However, the performance of EKF is sensitive to the signal properties, as shown in 7(c) and (d). While the test signal in 7(c) is similar to the signal in (a), except for the phase, the estimated parameters does not converge well (Fig. 7(d)) and the curve fitting fails (Fig. 7(c)). To assure a stable estimation result, we used the combination of the DFT and the RLS in this paper.

IV. SYSTEM IMPLEMENTATION

The algorithms described in the previous section are implemented as shown in the flowchart in Fig. 8. The main part of the software consists of five blocks: the frequency estimator block, the batch least squares (LS) block, the RLS block, the trajectory generator block, and a verification block. The motion of a human welder is first provided to the frequency estimator block. This block stores a sequence of

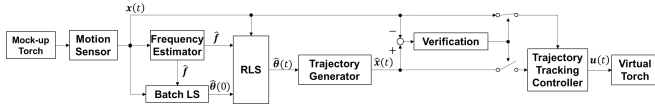


Fig. 8. Data flow diagram of the virtual experimental platform.

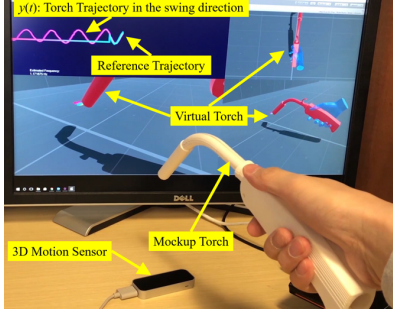


Fig. 9. The experimental platform. The teleoperation system is implemented in a virtual environment developed by Unity [13].

measured motion $x(t)$ in a buffer and estimates the frequency \hat{f} . The value of \hat{f} is sent to the batch LS and the RLS blocks. The batch LS block computes the initial values of the parameters $\hat{\theta}(0)$ by using $x(t)$ and \hat{f} . \hat{f} and $\hat{\theta}(0)$ are given to the RLS block. After these initial values are computed, the RLS block recursively analyzes the motion tracking data $x(t)$ streamed from a motion capture sensor. The result $\hat{x}(t)$ is then evaluated at the verification block. The difference between the trajectory generated with $\hat{x}(t)$ and a new time sequence of motion data is evaluated, and the estimate is validated when the squared prediction error becomes lower than a cutoff threshold and the estimated parameters converge to a constant value. The robot controller outputs the control input $u(t)$ to replace the human-generated motion command.

The above computation and control software was installed on the experimental platform shown in Fig. 9. A human operator moves a mock-up welding torch, which is measured with Leap Motion [14] connected to a computer via USB. A virtual torch, which is a torch model in a three-dimensional virtual space, moves according to the mock-up torch movement as a master-slave teleoperation system. A 3D human hand model is also displayed as an indicator of manual operation mode. This virtual hand disappears when the autopilot mode enables, which signifies that the control input has been switched from the manual mode, with the mockup torch, to the autopilot mode, with the reference trajectory generated using the estimated parameters. In the autopilot mode, the virtual torch follows the reference trajectory by PID control. The user can observe the virtual torch movement from three different viewpoints displayed on a monitor. The trajectories of the mock-up torch and the virtual torch are displayed on a window, in order to give the user a visual feedback of the torch movement and the estimation result.

V. EXPERIMENTAL RESULTS

A. Test with Synthetic Data

Prior to hand motion data, the developed algorithms were tested with a two-dimensional synthetic signal. The signal simulates a convex path, and consists of two functions with random noises: $x(t) = 0.6 \sin(2\pi \cdot 2.0 \cdot t + \pi/2) + 3.0t + 1.0$ and $y(t) = 18 \sin(2\pi \cdot 1.0 \cdot t + \pi)$. The window length of DFT was chosen as 3.0 seconds. Frequencies f_x and f_y estimated by DFT were 1.97 Hz and 1.00 Hz respectively. f_x was replaced with $2f_y = 2.00$ Hz.

Fig. 10 shows estimation results. Fig. 10(a) shows the trajectory of the test signal in the travel direction $x(t)$, and the estimated trajectory $\hat{x}(t)$. The test signal was streamed to the algorithm and the first 3.0 seconds of the data were stored in the buffer for the DFT. After the frequency was computed, the RLS parameter estimation started. The trends of the estimated parameters during the RLS estimation are shown in (b). The convergence criterion is that the changes in all the parameters are lower than 1%. The convergence time was 0.5 seconds. Then the reference trajectory, i.e. the estimated trajectory was generated as a red curved line in (a). Fig. 10(c) and (d) are about the y direction as well. Note that the horizontal axes are defined with 0 seconds as the RLS starting time. Fig. 10(e) shows a path formed by $\hat{x}(t)$ and $\hat{y}(t)$ as a red curved line. The computed path fits the test signal path shown as a blue curved line.

B. Test with Hand Motion Data

By using the developed virtual system, we demonstrated the real-time analysis of the motion tracking data and the seamless control transition. In the virtual environment, we assumed a situation in which the task executed by the robot is a flat welding, and the torch movement in vertical direction (z direction) is constrained by a guiding system (e.g. a mechanical joint tracking mechanism or a sensor-based tracking system). Because of this constraint, the torch height is constant during the welding task, and the torch movement is considered as a two-dimensional motion. The DFT time window was set as 5 seconds in this test. Frequencies f_x and f_y estimated by DFT were 1.56 Hz and 0.79 Hz respectively. f_x was replaced with $2f_y = 1.58$ Hz.

Fig. 11 shows estimation results. The projected reference trajectory (the red curve) generated by the estimation results connected smoothly with the human hand motion (the blue curve) as shown in Fig. 11(a) and (e). The parameter estimation trends are shown in (b) and (f), and the convergence time was 1.3 seconds. The trajectory, generated recursively at each sampling step until the validation, is shown in (c) and (g). (d) and (h) show the squared errors; their cutoff threshold was set as 1. In (b)-(d) and (f)-(h), the horizontal axes were defined such that the RLS starting time was 0 seconds. Both squared errors were lower than the threshold until 0.4 seconds, but the parameters had not converged yet. From 0.7 seconds to 1.1 seconds, the squared error of x became larger than the threshold (See (d)) because of the distortion of $x(t)$ (See (c)). At 1.2 seconds, the changes in parameters became lower than

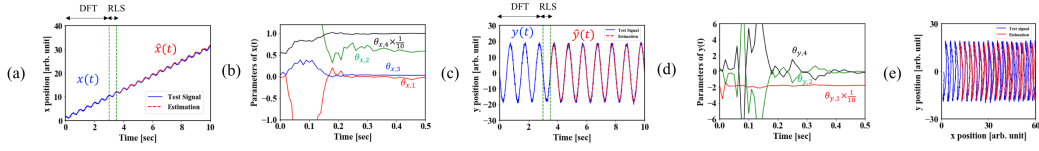


Fig. 10. Estimation results of synthetic signal. (a)The trajectory of the test signal in the travel direction $x(t)$, and the estimated trajectory $\hat{x}(t)$. (b)The trends of the estimated parameters of $x(t)$. (c)The trajectory of the test signal in the travel direction $y(t)$, and the estimated trajectory $\hat{y}(t)$. (d)The trends of the estimated parameters of $y(t)$. (e)A path formed by $\hat{x}(t)$ and $\hat{y}(t)$ as a red curved line.

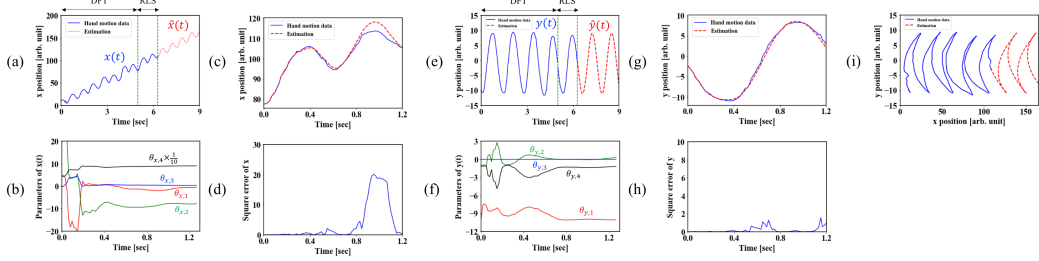


Fig. 11. Estimation results of hand motion data measured by the Leap Motion. (a)The trajectory of the test signal in the travel direction $x(t)$, and the estimated trajectory $\hat{x}(t)$. (b)The trends of the estimated parameters of $x(t)$ during the RLS estimation. (c)The trajectory computed recursively at each sampling step and the human motion in the x direction during the RLS. (d)The trajectory of the test signal in the travel direction $y(t)$, and the estimated trajectory $\hat{y}(t)$. (e)The trajectory computed recursively at each sampling step and the human motion in the y direction during the RLS. (f)The trends of the estimated parameters of $y(t)$ during the RLS estimation. (g)The trajectory computed recursively at each sampling step and the human motion in the y direction during the RLS. (h)The squared error $(\hat{y}_i - y_i)^2$. (i)A path formed by $\hat{x}(t)$ and $\hat{y}(t)$ shown as a red curved line.

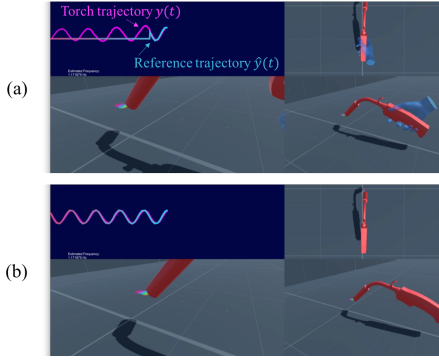


Fig. 12. (a)The virtual environment screen at an instant immediately after the control transition. (b)After the transition, the virtual torch works alone automatically.

1% and the squared errors became lower than the threshold again. Both validation criteria were satisfied at this point, so the control transition was then executed. (i) shows the human motion (the blue curve) and the reproduced path (the red curve) and they are smoothly connected. Fig. 12(a) shows the virtual environment screen at an instant immediately after the control transition. The virtual torch trajectory $y(t)$ seamlessly is following the reference trajectory $\hat{y}(t)$ by PID control. The manual mode indicator (the blue hand model) is disappearing in order to inform the user that the control transition has just been executed. After the transition, the virtual torch works alone automatically (Fig. 12(b)).

VI. CONCLUSIONS

In this paper, we presented a novel control transition method from a human demonstration to autopilot for welding robots as a new scheme of human and machine collaboration.

The human welder can start the welding task manually by directly manipulating the torch or using the teleoperation system with a hand motion capture device. This manual operation process is a robot teaching process as well. This means that the welder can define the robot torch movement intuitively using his/her motion. We modeled continuous welding movement as a combination of a sinusoidal function and a linear function, and demonstrated that this model can form typical welding movement patterns. The DFT and RLS algorithms were tested to estimate unknown parameters of the defined model in two-dimensional cases. We discussed that the ratio of the frequencies f_x and f_y have to satisfy the particular condition, otherwise, the welding path shape will collapse. Our RLS-based algorithm was tested with a synthetic path signal and actual human motion data, and it was confirmed that the control transition can be achieved within several seconds. As the future direction of development, we will extend our method to other types of welding patterns and three-dimensional reference trajectory generation. A verification and approval process before the control transition based on the welder's judgment about the quality of the parameter estimation should also be developed. Also, the control transition should be bi-directional between human and robot, in order for the welder to modify the robot's movement when the robot needs help. Our proposed method will effectively contribute to increasing the work productivity in welding because the welder can skip the teaching step that is usually necessary. It can also be utilized for other manufacturing processes that consist of repetitive motion patterns of the end effector, such as painting, gouging, gluing, cutting, and grinding.

REFERENCES

- [1] D. Lee, "Robots in the shipbuilding industry," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 5, pp. 442 – 450, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0736584514000131>
- [2] A. G. Dharmawan, A. A. Vibhute, S. Foong, G. S. Soh, and K. Otto, "A survey of platform designs for portable robotic welding in large scale structures," in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, Dec 2014, pp. 1683–1688.
- [3] C. Doumanidis and D. Hardt, "A model for in-process control of thermal properties during welding," *Journal of dynamic systems, measurement, and control*, vol. 111, no. 1, pp. 40–50, 1989.
- [4] A. Tarn and D. Hardt, "Weld pool impedance for pool geometry measurement: stationary and non-stationary pools," *Journal of dynamic systems, measurement, and control*, vol. 111, p. 545, 1989.
- [5] C. Doumanidis and D. E. Hardt, "Multivariable adaptive control of thermal properties during welding," *Journal of dynamic systems, measurement, and control*, vol. 113, no. 1, pp. 82–92, 1991.
- [6] A. Suzuki, D. Hardt, and L. Valavani, "Application of adaptive control theory to on-line gta weld geometry regulation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 113, no. 1, pp. 93–103, 1991.
- [7] J. Song and D. Hardt, "Closed-loop control of weld pool depth using a thermally based depth estimator," *WELDING JOURNAL-NEW YORK*, vol. 72, pp. 471–s, 1993.
- [8] D. Ni, A. W. W. Yew, S. K. Ong, and A. Y. C. Nee, "Haptic and visual augmented reality interface for programming welding robots," *Advances in Manufacturing*, vol. 5, no. 3, pp. 191–198, Sep 2017. [Online]. Available: <https://doi.org/10.1007/s40436-017-0184-7>
- [9] Y. Liu and Y. Zhang, "Toward welding robot with human knowledge: A remotely-controlled approach," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 769–774, April 2015.
- [10] S. J. Chen, N. Huang, Y. K. Liu, and Y. M. Zhang, "Machine assisted manual torch operation: system design, response modeling, and speed control," *Journal of Intelligent Manufacturing*, vol. 28, no. 6, pp. 1249–1258, Aug 2017. [Online]. Available: <https://doi.org/10.1007/s10845-015-1047-3>
- [11] M. Ramacciotti, M. Milazzo, F. Leoni, S. Roccella, and C. Stefanini, "A novel shared control algorithm for industrial robots," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, p. 1729881416682701, 2016. [Online]. Available: <https://doi.org/10.1177/1729881416682701>
- [12] Y. Liu and Y. Zhang, "Human welder 3-d hand movement learning in virtualized GTAW: Theory and experiments," in *Transactions on Intelligent Welding Manufacturing*, S. Chen, Y. Zhang, and Z. Feng, Eds. Singapore: Springer Singapore, 2019, pp. 3–25.
- [13] Unity. Accessed: 2018-09-01. [Online]. Available: <https://unity3d.com/>
- [14] Leap motion. Accessed: 2018-09-01. [Online]. Available: <https://www.leapmotion.com/>