

# BÁO CÁO NGHIÊN CỨU KỸ THUẬT: THIẾT KẾ VÀ TRIỂN KHAI MODULE WELDING SEQUENCER CHO HỆ THỐNG ROBOT HÀN CÔNG NGHIỆP

## 1. TỔNG QUAN VÀ PHẠM VI DỰ ÁN

### 1.1. Giới thiệu

Trong kỷ nguyên công nghiệp 4.0, tự động hóa quá trình hàn (Robotic Arc Welding) đóng vai trò then chốt trong việc đảm bảo năng suất và chất lượng sản phẩm cơ khí. Tuy nhiên, sự thành công của một hệ thống hàn robot không chỉ phụ thuộc vào cơ khí chính xác của cánh tay robot hay công nghệ biến tần của nguồn hàn, mà còn phụ thuộc vào "bộ não" điều phối giữa hai thực thể này: Module Trình tự Hàn (Welding Sequencer).

Báo cáo này được biên soạn dưới góc độ của một Kỹ sư Công nghệ Hàn (Welding Process Engineer), nhằm đáp ứng yêu cầu thiết kế một module điều khiển cho hệ thống robot sử dụng nguồn hàn MIG/MAG kỹ thuật số. Yêu cầu cốt lõi của dự án là robot phải đóng vai trò "Master" (Chủ) trong việc quản lý trình tự thời gian (Timing Sequence) và logic vận hành, trong khi nguồn hàn đóng vai trò "Slave" (Tớ) chịu trách nhiệm tính toán đường đặc tính Synergic (điện áp/dòng điện) dựa trên các setpoint được gửi xuống.

Mục tiêu tối thượng là xây dựng một kiến trúc phần mềm và phần cứng đảm bảo quy trình: **Mở khí** → **Chờ (Pre-flow)** → **Bật lửa** → **Chờ tín hiệu Arc OK** → **Bắt đầu di chuyển** diễn ra chính xác, tin cậy và an toàn.

### 1.2. Định nghĩa Vấn đề và Yêu cầu Kỹ thuật

Hệ thống hàn MIG/MAG (Metal Inert Gas / Metal Active Gas) là một quá trình phức tạp liên quan đến sự phối hợp giữa hồ quang điện, chuyển dịch kim loại lỏng, và khí bảo vệ. Một trong những nguyên nhân hàng đầu gây ra lỗi hàn trong môi trường robot (như rỗ khí, không ngấu, cháy ngược) là do sự sai lệch về thời gian (timing mismatch) giữa chuyển động của robot và trạng thái của hồ quang.<sup>1</sup>

Yêu cầu cụ thể từ người dùng nhấn mạnh vào việc kiểm soát trình tự thời gian chính xác mà không can thiệp vào thuật toán vật lý hàn (Synergic). Điều này đặt ra các thách thức kỹ thuật sau:

- Độ trễ tín hiệu (Signal Latency):** Phải xử lý độ trễ giữa lệnh Digital Output (DO) của robot và phản hồi Digital Input (DI) từ nguồn hàn.

2. **Đồng bộ hóa chuyển động (Motion Synchronization):** Robot tuyệt đối không được di chuyển khi hồ quang chưa hình thành ổn định ("Air Welding").
3. **Quản lý trạng thái (State Management):** Hệ thống phải hoạt động như một Máy trạng thái hữu hạn (Finite State Machine - FSM) để ngăn chặn các chuyển đổi trạng thái không hợp lệ.
4. **Tích hợp phần cứng:** Giao tiếp qua Digital I/O và Analog (0-10V) đòi hỏi các biện pháp chống nhiễu và chuẩn hóa tín hiệu nghiêm ngặt.

## 2. CƠ SỞ LÝ THUYẾT VỀ QUY TRÌNH HÀN GMAW ROBOT

Để thiết kế module WeldingSequencer hiệu quả, cần hiểu sâu sắc các hiện tượng vật lý diễn ra trong từng mili-giây của quá trình khởi tạo hồ quang.

### 2.1. Vật lý của Khởi tạo Hồ quang (Arc Initiation Physics)

Quá trình khởi tạo hồ quang không diễn ra tức thời. Khi robot gửi lệnh ARC\_START, một chuỗi các sự kiện vật lý phải xảy ra:

1. **Cấp dây (Wire Run-in):** Dây hàn được đẩy ra từ súng hàn với tốc độ chậm (Run-in speed) để tiếp xúc với phôi. Nếu tốc độ này quá cao, dây sẽ đâm vào phôi gây nổ (blast); nếu quá thấp, hồ quang sẽ chập chờn.<sup>3</sup>
2. **Ngắn mạch (Short Circuit):** Khi dây chạm phôi, điện áp giảm về gần 0V và dòng điện tăng vọt (dòng ngắn mạch).
3. **Ion hóa và Bốc bay (Ionization & Vaporization):** Dòng điện lớn làm nóng chảy và bốc hơi đoạn dây tiếp xúc, tạo ra cầu nối plasma dẫn điện (hồ quang).
4. **Ổn định (Stabilization):** Hệ thống điều khiển của nguồn hàn điều chỉnh dòng/áp để duy trì cột hồ quang ổn định.

Module WeldingSequencer phải cung cấp đủ thời gian cho các bước này diễn ra trước khi cho phép robot di chuyển. Nếu robot di chuyển trong giai đoạn ngắn mạch (bước 2), mối hàn sẽ bị khuyết tật "Cold Lap" (chồng mép hàn không ngấu) tại điểm bắt đầu.<sup>4</sup>

### 2.2. Vai trò của Khí bảo vệ và Pre-flow

Khí bảo vệ (như Argon,  $CO_2$ , hoặc hỗn hợp) có hai chức năng: bảo vệ kim loại nóng chảy khỏi Oxy/Nitơ trong không khí và hỗ trợ mỗi hồ quang.

- **Ion hóa:** Argon có thế năng ion hóa thấp hơn không khí, giúp hồ quang dễ mồi hơn.
- **Lưu lượng:** Cần một khoảng thời gian (Pre-flow time) để khí đẩy hết không khí trong ống dẫn (torch cable) ra ngoài. Nếu mồi hồ quang ngay khi mở van khí, đoạn khí đầu tiên phun ra vẫn lẫn không khí, gây rỗ khí (porosity) ngay tại điểm xuất phát.<sup>6</sup>

2.3. Hiện tượng Burnback và Crater Fill

Ở cuối đường hàn, nếu cắt điện và dừng dây cùng lúc, dây hàn đang nóng chảy có xu hướng dính chặt vào vũng hàn đang đông đặc hoặc dính ngược vào béc hàn (contact tip).

- **Crater Fill:** Giảm dòng điện từ từ để lấp đầy lõm cuối đường hàn, ngăn nứt chân chim.
- **Burnback:** Duy trì điện áp trong vài mili-giây sau khi ngừng cấp dây để đốt cháy đoạn dây thừa, đảm bảo dây không dính vào phôi và có độ thò (stick-out) chuẩn cho lần mỗi sau.<sup>8</sup>

Dù nguồn hàn hiện đại thường có chức năng Burnback nội bộ, WeldingSequencer của robot vẫn cần quản lý trạng thái này để đảm bảo robot không rút súng hàn ra khỏi vùng khí bảo vệ quá sớm.

3. THIẾT KẾ KIẾN TRÚC HỆ THỐNG (SYSTEM ARCHITECTURE)

3.1. Mô hình Phân cấp Master-Slave

Hệ thống được thiết kế theo mô hình phân cấp, trong đó Robot Controller giữ vai trò điều phối cấp cao.

- **Robot Controller (Sequencer):** Quản lý "KHI NÀO" (When) và "NẾU" (If). Ví dụ: "Khi nào thì mở khí?", "Nếu không có hồ quang thì làm gì?".
- **Power Source (Synergic Core):** Quản lý "NHƯ THẾ NÀO" (How). Ví dụ: "Với 200A, điện áp phải là bao nhiêu V để hồ quang ổn định?". Nguồn hàn tự động tra bảng Synergic Curve để điều chỉnh.<sup>1</sup>

3.2. Giao diện Phần cứng (Hardware Interface Layer - HAL)

Để thực hiện quy trình yêu cầu, chúng ta cần xác định danh sách tín hiệu I/O (I/O Map) chính xác. Dựa trên các tài liệu kỹ thuật từ Fronius, Miller, và Panasonic <sup>1</sup>, bảng dưới đây mô tả cấu hình I/O tiêu chuẩn cho module này.

3.2.1. Digital Outputs (Robot → Nguồn hàn)

Tên Tín hiệu (Code)	Loại	Chức năng	Mức Logic	Ghi chú Kỹ thuật
DO_GAS_VALVE	Digital	Điều khiển van khí điện từ (Solenoid).	High = Open	Cần tách biệt với Arc Start để thực hiện

				Pre-flow độc lập. <sup>12</sup>
<b>DO_ARC_START</b>	Digital	Lệnh khởi động hàn (Main Contactor).	High = Weld	Kích hoạt chu trình nội bộ của nguồn hàn (cấp dây, đóng điện).
<b>DO_WIRE_INCREMENT</b>	Digital	Đẩy dây nhanh (không hàn).	High = Feed	Dừng khi thay cuộn dây hoặc kiểm tra.
<b>DO_WIRE_RETRACT</b>	Digital	Rút dây.	High = Retract	Hữu ích để xử lý sự cố dính dây hoặc căn chỉnh Stick-out.
<b>DO_ERROR_RESET</b>	Digital	Xóa lỗi nguồn hàn.	Pulse (High → Low)	Dùng để reset sau khi khắc phục sự cố (ví dụ: quá nhiệt).
<b>DO_JOB_BIT0..3</b>	Digital	Chọn chương trình hàn (Job Select).	Binary/BCD	(Tùy chọn) Để robot chọn chế độ Synergic (Vật liệu/Khí/Đường kính).

### 3.2.2. Digital Inputs (Nguồn hàn → Robot)

Tên Tín hiệu (Code)	Loại	Chức năng	Mức Logic	Ghi chú Kỹ thuật
<b>DI_ARC_OK</b>	Digital	Xác nhận hồ quang ổn định (Current	High = OK	<b>Tín hiệu quan trọng nhất.</b> Cho phép

		Detect).		robot bắt đầu di chuyển. <sup>14</sup>
<b>DI_READY</b>	Digital	Nguồn hàn sẵn sàng.	High = Ready	Nguồn đã bật, không lỗi, sẵn sàng nhận lệnh.
<b>DI_WELD_ERROR</b>	Digital	Báo lỗi chung.	High = Fault	Quá nhiệt, mất pha, hết nước làm mát. <sup>1</sup>
<b>DI_WIRE_STUCK</b>	Digital	Báo dính dây (Anti-stick).	High = Stuck	Dây hàn dính vào béc hàn hoặc phôi sau khi tắt hồ quang.
<b>DI_GAS_FLOW</b>	Digital	Cảm biến dòng khí (Flow sensor).	High = Flowing	(Khuyến nghị) Feedback thực tế dòng khí thay vì chỉ tin vào van điện từ.

### 3.2.3. Analog Interface (Điều khiển Tham số)

Mặc dù sử dụng chế độ Synergic, robot vẫn cần gửi tín hiệu tham chiếu để điều chỉnh năng lượng hàn.

- **AO\_WFS\_REF (0-10V):** Tham chiếu tốc độ cấp dây (Wire Feed Speed) hoặc Dòng điện (Amperage). Trong chế độ Synergic, tín hiệu này thường đại diện cho "Mức năng lượng" (Power Level). Ví dụ: 5V = 150A.<sup>1</sup>
- **AO\_VOLT\_REF (0-10V):** Tham chiếu Điện áp hoặc Arc Length Correction (Trim). Trong Synergic, 5V thường là mức chuẩn (0 trim), <5V là ngắn hồ quang, >5V là dài hồ quang.<sup>4</sup>

### 3.3. Sơ đồ Kết nối và Cách ly

Môi trường hàn hồ quang tạo ra nhiễu điện từ (EMI) cực lớn do dòng điện cao tần và biến thiên dòng lớn (di/dt).

- **Cách ly quang (Opto-isolation):** Tất cả các tín hiệu Digital I/O giữa robot và nguồn hàn *bắt buộc* phải đi qua bộ cách ly quang hoặc Relay trung gian để bảo vệ vi xử lý của robot khỏi xung điện áp cao.<sup>1</sup>

- **Cáp Analog:** Sử dụng cáp xoắn đôi có bọc kim (Shielded Twisted Pair), với vỏ bọc nối đất *chỉ tại một đầu* (thường là phía tủ điều khiển robot) để tránh vòng lặp đất (Ground Loop) gây sai lệch tín hiệu điều khiển 0-10V.<sup>10</sup>

## 4. THIẾT KẾ CHI TIẾT MODULE WELDING SEQUENCER

Module WeldingSequencer được thiết kế dưới dạng một **Máy trạng thái hữu hạn (Finite State Machine - FSM)**. Mô hình này đảm bảo tính tất định (determinism), nghĩa là với cùng một đầu vào và trạng thái hiện tại, hệ thống luôn chuyển sang một trạng thái tiếp theo xác định, loại bỏ các hành vi ngẫu nhiên nguy hiểm.<sup>16</sup>

### 4.1. Định nghĩa Các Trạng thái (States)

Dựa trên quy trình yêu cầu, chúng ta định nghĩa 8 trạng thái cơ bản cho FSM:

1. **STATE\_IDLE (0):** Hệ thống nghỉ, chờ lệnh CMD\_START. Tất cả đầu ra tắt (trừ tín hiệu Ready).
2. **STATE\_PREFLOW (1):** Mở van khí, chờ thời gian  $T_{Pre}$ .
3. **STATE\_IGNITION (2):** Bật DO\_ARC\_START, chờ tín hiệu DI\_ARC\_OK. Đây là giai đoạn mồi lửa.
4. **STATE\_STABILIZE (3):** Hồ quang đã cháy, giữ yên robot trong thời gian  $T_{Stab}$  để ổn định vùng hàn.
5. **STATE\_WELD (4):** Hồ quang ổn định, gửi cờ MOTION\_ENABLE cho robot di chuyển.
6. **STATE\_CRATER (5):** Nhận lệnh dừng, chuyển sang thông số Crater, giảm tốc độ.
7. **STATE\_BURNBACK (6):** Tắt dây, giữ điện áp trong thời gian  $T_{Burn}$  để chống dính.
8. **STATE\_POSTFLOW (7):** Tắt hồ quang, giữ khí bảo vệ trong thời gian  $T_{Post}$ .
9. **STATE\_FAULT (99):** Trạng thái lỗi (Mất hồ quang, dính dây, lỗi nguồn).

### 4.2. Biểu đồ Thời gian và Logic Chuyển đổi (Timing Diagram & Transition Logic)

Dưới đây là phân tích chi tiết từng bước chuyển đổi, tích hợp các tham số thời gian và tín hiệu I/O.

#### Giai đoạn 1: Khởi động (Start Sequence)

- **Từ IDLE sang PREFLOW:**
  - *Điều kiện:* Nhận lệnh CMD\_START = TRUE.
  - *Hành động:* DO\_GAS\_VALVE = ON. Khởi động Timer  $t_{state}$ .
  - *Logic:* Tại sao phải chờ Pre-flow? Nếu ống dẫn khí dài 10m, đường kính trong 6mm, thể tích khí cần đẩy là  $V = \pi \times (0.3)^2 \times 1000 \approx 280cm^3$ . Với lưu lượng 15

L/min ( $250 \text{ cm}^3/\text{s}$ ), cần ít nhất 1.2 giây để khí mới đến súng. Tuy nhiên, trong thực tế sản xuất, ống luôn chứa sẵn khí (chỉ bị lẫn khí ở đầu vòi), nên  $T_{Pre}$  thường đặt 0.1s - 0.5s.<sup>6</sup>

- Tham số: Param\_PreFlowTime (Ví dụ: 0.3s).
- **Từ PREFLOW sang IGNITION:**
  - Điều kiện:  $t\_state \geq \text{Param\_PreFlowTime}$ .
  - Hành động: DO\_ARC\_START = ON. Thiết lập Analog Setpoint = Param\_Start\_WFS / Param\_Start\_Volts.
  - Hành động phụ: Reset Watchdog Timer cho việc mỗi hồ quang.
  - Lưu ý: Robot vẫn đứng yên.
- **Từ IGNITION sang STABILIZE:**
  - Điều kiện:  $DI\_ARC\_OK == \text{HIGH}$ .
  - Hành động: Chuyển trạng thái. Bắt đầu Timer  $t\_stab$ .
  - Xử lý lỗi: Nếu  $t\_state > \text{Param\_IgnitionTimeout}$  (ví dụ 3.0s) mà chưa có  $DI\_ARC\_OK$ , chuyển sang STATE\_FAULT với mã lỗi "Ignition Failure".<sup>18</sup>
- **Từ STABILIZE sang WELD:**
  - Điều kiện:  $t\_stab \geq \text{Param\_StabilizeTime}$  (Ví dụ: 0.2s).
  - Hành động: Gửi tín hiệu MOTION\_PERMIT = TRUE tới bộ điều khiển chuyển động của robot. Chuyển Analog Setpoint sang Param\_Weld\_WFS / Param\_Weld\_Volts.
  - Ý nghĩa: Thời gian ổn định (Stabilize) giúp tạo vũng hàn ban đầu (weld pool) đủ nhiệt trước khi di chuyển, tránh khuyết tật không ngẫu đầu đường hàn.

## Giai đoạn 2: Hàn (Welding Process)

- **Trong STATE\_WELD:**
  - Giám sát: Liên tục kiểm tra  $DI\_ARC\_OK$ .
  - Sự cố: Nếu  $DI\_ARC\_OK$  chuyển xuống LOW (mất hồ quang đột ngột):
    1. Ngắt MOTION\_PERMIT ngay lập tức (Robot dừng).
    2. Thử kích hoạt lại (Re-strike) trong 200ms.
    3. Nếu không thành công, chuyển sang STATE\_FAULT.<sup>19</sup>

## Giai đoạn 3: Kết thúc (End Sequence)

- **Từ WELD sang CRATER:**
  - Điều kiện: Nhận lệnh CMD\_STOP hoặc đi hết quỹ đạo hàn.
  - Hành động: Robot dừng hoặc giảm tốc độ di chuyển. Chuyển Analog Setpoint sang Param\_Crater\_WFS (thường 60-80% trị số hàn) và Param\_Crater\_Volts. Giữ trạng thái này trong Param\_CraterTime.<sup>8</sup>
  - Mục đích: Giảm dòng từ từ giúp khí thoát khỏi vũng hàn đang đông đặc, ngăn chặn vết nứt lõm (crater crack).
- **Từ CRATER sang BURNBACK:**
  - Điều kiện:  $t\_state \geq \text{Param\_CraterTime}$ .
  - Hành động: Tắt dây hàn (Analog WFS = 0 hoặc DO\_STOP\_WIRE nếu có). **Vẫn giữ**

**DO\_ARC\_START = ON.**

- *Logic Burnback*: Đây là điểm mâu thuẫn thường gặp. Một số nguồn hàn tự quản lý Burnback. Nếu nguồn hàn tự quản lý, robot chỉ cần tắt DO\_ARC\_START và nguồn hàn sẽ tự làm phần còn lại. Tuy nhiên, với yêu cầu "Kiểm soát chính xác thời gian" từ người dùng, Sequencer nên hỗ trợ cả hai chế độ. Ở đây giả định chế độ Robot điều khiển: Robot giữ Contactor đóng nhưng ngừng cấp dây trong Param\_BurnbackTime (ví dụ 0.05s - 0.1s) để hồ quang đốt ngắn dây lại.<sup>9</sup>
- *Kết thúc Burnback*: DO\_ARC\_START = OFF.
- **Từ BURNBACK sang POSTFLOW:**
  - *Điều kiện*: t\_state >= Param\_BurnbackTime.
  - *Hành động*: DO\_ARC\_START đã tắt. DO\_GAS\_VALVE vẫn giữ ON.
  - *Thời gian*: Chờ Param\_PostFlowTime (Ví dụ 1.0s - 3.0s).
  - *Mục đích*: Bảo vệ điện cực vonfram (TIG) hoặc dây hàn nóng đỏ (MIG) và vũng hàn khỏi bị oxy hóa khi nguội đi.<sup>20</sup>
- **Từ POSTFLOW về IDLE:**
  - *Hành động*: DO\_GAS\_VALVE = OFF. Reset toàn bộ biến nội bộ. Sẵn sàng cho chu trình mới.

## 5. TRIỂN KHAI KỸ THUẬT VÀ THUẬT TOÁN (IMPLEMENTATION & ALGORITHMS)

Phần này mô tả cấu trúc phần mềm bằng giả mã C++ (phổ biến trong phát triển controller robot) để minh họa logic của WeldingSequencer.

### 5.1. Cấu trúc Lớp (Class Structure)

C++

```
// Cấu trúc tham số hàn
struct WeldParameters {
    float preflow_time_sec;    // Thời gian khí trước (0.1 - 5.0s)
    float ignition_timeout_sec; // Thời gian chờ mỗi tối đa (1.0 - 5.0s)
    float stabilize_time_sec;  // Thời gian ổn định (0.0 - 2.0s)
    float crater_time_sec;     // Thời gian lấp rãnh (0.0 - 5.0s)
    float burnback_time_sec;   // Thời gian chống dính (0.0 - 0.5s)
    float postflow_time_sec;   // Thời gian khí sau (0.5 - 10.0s)

    // Analog Setpoints (0.0 - 10.0V)
    float weld_voltage_ref;
```



```

float weld_wfs_ref;
float crater_voltage_ref;
float crater_wfs_ref;
};

// Enum trạng thái
enum SequencerState {
    IDLE,
    PREFLOW,
    IGNITION,
    STABILIZE,
    WELD,
    CRATER,
    BURNBACK,
    POSTFLOW,
    FAULT
};

class WeldingSequencer {
private:
    SequencerState currentState;
    double timerStartTimestamp;
    bool arcSignalDebounced;

    // Các hàm nội bộ
    void setDigitalOutput(int pin, bool state);
    void setAnalogOutput(int channel, float voltage);
    bool readDigitalInput(int pin);
    double getElapsedTime();

public:
    void init();
    void cycleUpdate(bool cmdStart, bool cmdStop, WeldParameters params);
    bool isMotionPermitted(); // Hàm robot gọi để biết có được di chuyển không
};

```

## 5.2. Thuật toán Vòng lặp Điều khiển (Control Loop Logic)

Thuật toán này cần được gọi trong vòng lặp thời gian thực (Real-time loop) của robot, với chu kỳ quét khoảng 5ms - 10ms để đảm bảo độ chính xác.<sup>17</sup>

### Logic Xử lý Tín hiệu Arc OK (Debouncing):

Tín hiệu DI\_ARC\_OK thường rất nhiễu trong giai đoạn ngắn mạch (Short Circuit Transfer). Nếu

đọc thô (raw), FSM có thể nhảy trạng thái liên tục.

- *Giải pháp*: Sử dụng bộ đếm hoặc bộ lọc thời gian.
- *Quy tắc*: DI\_ARC\_OK chỉ được coi là HIGH nếu nó giữ mức 1 trong ít nhất 20ms liên tục. Ngược lại, chỉ coi là mất hồ quang nếu nó ở mức 0 trong >50ms (để bỏ qua các lần tắt ngắn mạch tự nhiên).<sup>14</sup>

### Chi tiết thuật toán chuyển trạng thái (Pseudocode):

C++

```
void WeldingSequencer::cycleUpdate(bool cmdStart, bool cmdStop, WeldParameters params) {  
    // 1. Đọc và lọc tín hiệu đầu vào  
    bool rawArcOK = readDigitalInput(DI_ARC_PIN);  
    this->arcSignalDebounced = debounceSignal(rawArcOK);  
  
    // 2. Máy trạng thái (FSM)  
    switch (currentState) {  
        case IDLE:  
            if (cmdStart) {  
                setDigitalOutput(DO_GAS_PIN, HIGH);  
                timerStartTimestamp = getCurrentTime();  
                currentState = PREFLOW;  
            }  
            break;  
  
        case PREFLOW:  
            if (getElapsedTime() >= params.preflow_time_sec) {  
                // Hết thời gian Pre-flow -> Bật lửa  
                setAnalogOutput(AO_WFS, params.weld_wfs_ref); // Set tham số hàn  
                setAnalogOutput(AO_VOLT, params.weld_voltage_ref);  
                setDigitalOutput(DO_ARC_START_PIN, HIGH);  
                timerStartTimestamp = getCurrentTime(); // Reset timer cho timeout  
                currentState = IGNITION;  
            }  
            break;  
  
        case IGNITION:  
            // Chờ tín hiệu Arc OK  
            if (this->arcSignalDebounced) {
```

```

        timerStartTimestamp = getCurrentTime();
        currentState = STABILIZE;
    }
    else if (getElapsedTime() > params.ignition_timeout_sec) {
        // Quá giờ mà không có lửa -> Lỗi
        currentState = FAULT;
        reportError("ARC START FAILURE");
    }
    break;

case STABILIZE:
    // Chờ ổn định vũng hàn
    if (!this->arcSignalDebounced) {
        currentState = IGNITION; // Mất lửa thì quay lại chờ (hoặc lỗi)
    }
    else if (getElapsedTime() >= params.stabilize_time_sec) {
        currentState = WELD; // Đủ thời gian ổn định -> Cho phép hàn
    }
    break;

case WELD:
    // Trạng thái hàn chính
    if (cmdStop) {
        // Nhận lệnh dừng -> Chuyển sang Crater
        setAnalogOutput(AO_WFS, params.crater_wfs_ref);
        setAnalogOutput(AO_VOLT, params.crater_voltage_ref);
        timerStartTimestamp = getCurrentTime();
        currentState = CRATER;
    }
    else if (!this->arcSignalDebounced) {
        // Mất lửa giữa chừng -> Dừng chuyển động, báo lỗi
        currentState = FAULT;
        reportError("PROCESS LOST");
    }
    break;

case CRATER:
    if (getElapsedTime() >= params.crater_time_sec) {
        setDigitalOutput(DO_ARC_START_PIN, LOW); // Tắt hồ quang
        // Lưu ý: Nếu dùng Robot Burnback, logic sẽ phức tạp hơn ở đây
        timerStartTimestamp = getCurrentTime();
        currentState = BURNBACK; // Hoặc nhảy thẳng Post-flow tùy cấu hình
    }

```

```
break;
```

```
//... Các trạng thái Burnback và Postflow tương tự...
```

```
}  
}
```

## 6. XỬ LÝ LỖI VÀ PHỤC HỒI (ERROR HANDLING & RECOVERY)

Một hệ thống tự động tốt được đánh giá qua khả năng xử lý sự cố chứ không chỉ là chạy trơn tru khi mọi thứ hoàn hảo. Dưới đây là các chiến lược phục hồi cho các lỗi phổ biến được trích xuất từ kinh nghiệm vận hành thực tế.<sup>18</sup>

### 6.1. Lỗi Mồi Hồ quang (Ignition Failure)

**Hiện tượng:** Robot gửi lệnh ARC\_START, dây chạy ra, chạm phôi nhưng không bắt lửa, hoặc cháy vệt rồi tắt.

**Nguyên nhân:** Bề mặt phôi bẩn/sơn, tiếp xúc mass kém, dây hàn bị cắt cụt (không nhọn).

**Chiến lược Phục hồi (Auto-Recovery Logic):**

1. **Phát hiện:** Ignition Timeout kích hoạt.
2. **Hành động tức thời:** Ngắt DO\_ARC\_START, Ngắt DO\_GAS.
3. **Quy trình Thử lại (Retry Routine):**
  - Kích hoạt DO\_WIRE\_RETRACT trong 0.5s để rút dây về (tránh dây đâm cong vào phôi).
  - Nâng robot lên trục Z (+10mm).
  - Di chuyển robot lùi lại một khoảng nhỏ (-5mm dọc đường hàn).
  - Thử lại quy trình khởi động (tối đa 3 lần).
  - Nếu vẫn thất bại: Dừng hệ thống và báo đèn đỏ cho người vận hành.

### 6.2. Lỗi Dính dây (Wire Stick / Burnback Error)

**Hiện tượng:** Sau khi kết thúc hàn, dây hàn dính chặt vào vũng hàn đông đặc. Robot không thể di chuyển đi chỗ khác.

**Nguy hiểm:** Nếu robot cố di chuyển, súng hàn sẽ bị bẻ cong, hoặc trục robot bị quá tải (Servo Overload).

**Giải pháp:**

- Sử dụng tín hiệu DI\_WIRE\_STUCK từ nguồn hàn (nếu có).
- Nếu không có tín hiệu này, logic Sequencer có thể kiểm tra dòng điện: Nếu DO\_ARC\_START đã tắt được 1s mà DI\_ARC\_OK (hoặc cảm biến dòng analog) vẫn báo có dòng điện ngắn mạch, nghĩa là dây đang dính.
- **Hành động:** Khóa chuyển động robot (MOTION\_INHIBIT). Gửi cảnh báo "WIRE STUCK".

Yêu cầu người vận hành cắt dây thủ công. Một số hệ thống cao cấp có thể thử phóng một xung điện mạnh (Pulse) để làm nóng chảy điểm dính.<sup>22</sup>

## 7. GIAO DIỆN ĐIỆN VÀ TỐI ƯU HÓA TÍN HIỆU

Để đảm bảo tính "chính xác về thời gian" như yêu cầu, phần cứng phải được thiết kế để giảm thiểu nhiễu và độ trễ.

### 7.1. Độ trễ Hệ thống (System Latency)

- Van khí:** Van solenoid tiêu chuẩn mất 20ms-50ms để mở hoàn toàn. Sequencer phải cộng thêm thời gian này vào Param\_PreFlowTime.
- Relay:** Relay cơ khí mất ~10ms. Nên sử dụng Relay bán dẫn (Solid State Relay - SSR) hoặc Optocoupler để có thời gian đáp ứng <1ms.
- Giao tiếp:** Nếu dùng I/O trực tiếp (Hardwired), độ trễ gần như bằng 0. Nếu dùng qua PLC trung gian (Robot → PLC → Nguồn hàn), độ trễ giao tiếp (Bus cycle time) có thể lên tới 10-20ms. Với yêu cầu chính xác cao, khuyến nghị đấu nối trực tiếp I/O từ Robot vào Nguồn hàn.

### 7.2. Bảng Dữ liệu Tối ưu hóa Tham số (Parameter Tuning Guide)

Bảng dưới đây cung cấp các giá trị tham chiếu để thiết lập ban đầu cho Sequencer, giúp Kỹ sư vận hành tiết kiệm thời gian thử nghiệm.<sup>3</sup>

Tham số	Giá trị Thép (Steel)	Giá trị Nhôm (Alu)	Giá trị Inox (SS)	Giải thích Tác động
Pre-flow	0.3s	0.5s	0.4s	Nhôm cần bảo vệ kỹ hơn do dễ bị oxy hóa nhanh.
Ignition Time	2.0s (Timeout)	2.0s	2.0s	Giới hạn an toàn.
Stabilize	0.2s	0.1s	0.2s	Thép cần thời gian nung nóng ban đầu lâu hơn nhôm.
Crater Time	0.5s	0.8s	0.6s	Nhôm có hệ số

				co ngót nhiệt lớn, cần Crater lâu để tránh nứt.
<b>Burnback</b>	0.08s	0.12s	0.08s	Dây nhôm mềm và dẫn nhiệt nhanh, dễ bị chảy ngược lên béc hàn, cần Burnback kỹ.
<b>Post-flow</b>	1.0s	3.0s	2.0s	Inox và Nhôm cần khí sau lâu để bề mặt mỗi hàn sáng bóng.

## 8. KẾT LUẬN

Việc thiết kế module WeldingSequencer cho hệ thống robot hàn MIG/MAG kỹ thuật số không chỉ đơn thuần là lập trình Bật/Tắt tín hiệu, mà là sự tổng hòa giữa kiến thức vật lý hàn, lý thuyết điều khiển tự động và kỹ thuật phần mềm thời gian thực.

Bản báo cáo này đã trình bày một giải pháp toàn diện, đáp ứng chính xác yêu cầu của người dùng về việc tách biệt quản lý trình tự (do Robot đảm nhiệm) và quản lý năng lượng (do Nguồn hàn đảm nhiệm). Với kiến trúc Máy trạng thái hữu hạn (FSM) được đề xuất, hệ thống đảm bảo:

1. **Chất lượng khởi đầu:** Loại bỏ rỉ khí nhờ logic Pre-flow chính xác.
2. **Độ tin cậy:** Ngăn chặn lỗi "Air Welding" nhờ cơ chế bắt tay DI\_ARC\_OK chặt chẽ.
3. **Tuổi thọ thiết bị:** Giảm thiểu dính dây và hỏng béc hàn nhờ quy trình Burnback/Crater hợp lý.
4. **An toàn:** Tích hợp các quy trình xử lý lỗi tự động.

Giải pháp này hoàn toàn khả thi để triển khai trên các bộ điều khiển robot công nghiệp phổ biến (như FANUC, Yaskawa, ABB, KUKA) sử dụng ngôn ngữ lập trình bậc cao hoặc logic PLC tích hợp. Đây là nền tảng vững chắc để xây dựng một cell hàn robot hiệu suất cao, sẵn sàng cho sản xuất hàng loạt.

### Works cited

1. Robot Interface - Migatronik, accessed February 1, 2026, [https://www.migatronik.com/media/1386/50173004\\_manual\\_robot-interface-mig.pdf](https://www.migatronik.com/media/1386/50173004_manual_robot-interface-mig.pdf)
2. Troubleshooting Robotic Welding Arc Faults - ABIBLOG, accessed February 1, 2026, <https://blog.binzel-abicor.com/usa/troubleshooting-robotic-welding-arc-faults>
3. Analog Interface, AD1359-1 | PDF | Welding | Construction - Scribd, accessed February 1, 2026, <https://www.scribd.com/document/292527006/Analog-Interface-AD1359-1>
4. Mr. Roboto: Welding fundamentals for managers - The Fabricator, accessed February 1, 2026, <https://www.thefabricator.com/thefabricator/article/automationrobotics/mr-robot-o-welding-fundamentals-for-managers>
5. Digital v 'old school' settings? | Page 2 | MIG Welding Forum, accessed February 1, 2026, <https://www.mig-welding.co.uk/forum/threads/digital-v-old-school-settings.97180/page-2>
6. MIG Welding Gas Pressure Settings (with Charts) - Weld Guru, accessed February 1, 2026, <https://weldguru.com/mig-welding-gas-pressure/>
7. How to Set Gas Pressure For MIG Welding With Charts And Explanations - YesWelder, accessed February 1, 2026, <https://yeswelder.com/blogs/yeswelder/how-to-set-gas-pressure-for-mig-welding-with-charts-and-explanations>
8. Welding Parameters Explained - YesWelder, accessed February 1, 2026, <https://yeswelder.com/blogs/yeswelder/welding-parameters-explained>
9. Burnback control | MIG Welding Forum, accessed February 1, 2026, <https://www.mig-welding.co.uk/forum/threads/burnback-control.46/>
10. Welding Burnback: What Is It & How to Adjust the Controls - UNIMIG, accessed February 1, 2026, <https://unimig.com.au/blog/welding-burnback-what-is-it-and-how-to-adjust-the-controls>
11. Adding Capability to Your Robotic Welding Process | Y-Blog - Yaskawa Motoman, accessed February 1, 2026, <https://www.motoman.com/en-us/about/blog/adding-capability-to-your-robotic-welding-process>
12. Robotic Interface II - Miller Welding, accessed February 1, 2026, [https://www.millerwelds.com/files/owners-manuals/o172324d\\_mil.pdf](https://www.millerwelds.com/files/owners-manuals/o172324d_mil.pdf)
13. YA-1NA\*\*\*/ YA-1PA\*\*\*, accessed February 1, 2026, <https://icdn.tradew.com/file/201606/1569362/pdf/7593147.pdf>
14. User Information - Fronius International, accessed February 1, 2026, <https://www.fronius.com/~/-downloads/Perfect%20Welding/User%20Information/42%2C0426%2C0227%2CEA.pdf>
15. analysis of welding parameters in gas metal arc welding by a welding robot - Middle East Technical University, accessed February 1, 2026, <https://open.metu.edu.tr/bitstream/handle/11511/15987/index.pdf>

16. Draw a finite state machine for a welding machine - Electronics Stack Exchange, accessed February 1, 2026,  
<https://electronics.stackexchange.com/questions/122737/draw-a-finite-state-machine-for-a-welding-machine>
17. Implementing a Real-Time State Machine in Modern C++ - honeytreeLabs, accessed February 1, 2026,  
<https://honeytreelabs.com/posts/real-time-state-machine-in-cpp/>
18. Improved Response to Welding Arc Failure - Beacon Platform - Hirebotics, accessed February 1, 2026,  
<https://help.hirebotics.com/en/articles/10080737-improved-response-to-welding-arc-failure>
19. Dealing with Arc Start Failures - Yaskawa Knowledge Center, accessed February 1, 2026,  
<https://knowledge.motoman.com/hc/en-us/articles/4408792362903-Dealing-with-Arc-Start-Failures>
20. OM-154 145B Robotic Interface Control Gas/Current Sensing Control - Miller Welding, accessed February 1, 2026,  
<https://www.millerwelds.com/files/owners-manuals/o154145b.pdf>
21. Basic Fault Recovery Guide | Robots.com, accessed February 1, 2026,  
<https://www.robots.com/articles/basic-fault-recovery-a-step-by-step-guide-for-successful-execution>
22. Solving Welding Burnbacks in Robotic Applications - ABIBLOG, accessed February 1, 2026,  
<https://blog.binzel-abicor.com/usa/solving-welding-burnbacks-in-robotic-applications>
23. LGA Interface - Lincoln Electric, accessed February 1, 2026,  
<https://assets.lincolnelectric.com/assets/EU/OperatorManuals/IM3056rev01-ENG.pdf>