

# Báo cáo Nghiên cứu Chuyên sâu: Phát triển Module WeldPatternGenerator cho Hệ thống Điều khiển Robot Hàn 6 Trục

## 1. Tổng quan và Kiến trúc Hệ thống Điều khiển

### 1.1. Giới thiệu

Trong lĩnh vực chế tạo robot công nghiệp, đặc biệt là robot hàn hồ quang (Arc Welding Robot), phần mềm điều khiển đóng vai trò cốt lõi trong việc định hình chất lượng sản phẩm. Trong khi các bộ nội suy chuyển động cơ bản (Linear/Joint Interpolation) giải quyết vấn đề đưa mỏ hàn từ điểm A đến điểm B, thì module tạo mẫu đường hàn (WeldPatternGenerator) mới chính là thành phần quyết định tính "nghệ thuật" và độ bền kết cấu của mối hàn. Yêu cầu phát triển một module có khả năng xử lý Weaving (dao động), Dwell Time (thời gian dừng), Stitch Welding (hàn đính), và Multi-pass (hàn nhiều lớp) đòi hỏi sự am hiểu sâu sắc về toán học không gian, động học robot và vật lý hồ quang.<sup>1</sup>

Báo cáo này cung cấp một phân tích toàn diện, đi sâu vào các công thức toán học, logic điều khiển và giả mã (pseudo-code) cần thiết để triển khai các kỹ thuật này bằng ngôn ngữ C++. Mục tiêu là cung cấp một tài liệu tham khảo kỹ thuật chi tiết, hỗ trợ việc chuyển đổi các lý thuyết hàn thành các thuật toán điều khiển thời gian thực chính xác.

### 1.2. Vai trò của WeldPatternGenerator trong Kiến trúc Điều khiển

Trong kiến trúc phần mềm điều khiển robot tiêu chuẩn, WeldPatternGenerator hoạt động như một lớp trung gian (middleware) giữa bộ lập kế hoạch quỹ đạo (Trajectory Planner) và bộ giải động học ngược (Inverse Kinematics Solver).

- **Đầu vào:** Quỹ đạo danh nghĩa (Nominal Path) được xác định bởi các điểm dạy (Teaching Points) bao gồm vị trí  $(x, y, z)$  và hướng ( $Roll, Pitch, Yaw$  hoặc Quaternion), cùng với các tham số công nghệ hàn (biên độ lắc, tần số, thời gian dừng).
- **Xử lý:** Module này tính toán các vector bù trừ (offset vectors) thời gian thực dựa trên hàm sóng (waveform function) và hệ tọa độ cục bộ tại đầu mỏ hàn (TCP - Tool Center Point).
- **Đầu ra:** Tọa độ TCP biến thiên theo thời gian ( $P_{mod}(t)$ ) được gửi đến bộ giải động học ngược để tính toán các góc khớp  $(q_1, \dots, q_6)$ .<sup>1</sup>

Sự tách biệt giữa "chuyển động mang" (carrier motion - di chuyển dọc theo đường hàn) và "chuyển động tạo hình" (pattern motion - dao động ngang) là nguyên tắc cơ bản để thiết kế module này. Điều này cho phép người vận hành thay đổi kiểu dao động mà không cần dạy lại

đường đi cơ bản.

## 2. Cơ sở Toán học: Hệ Tọa độ và Khung Dệt (Weaving Frame)

Để thực hiện bất kỳ kỹ thuật Weaving nào, trước hết ta phải xác định được một hệ tọa độ cục bộ di chuyển dọc theo đường hàn. Hệ tọa độ này xác định phương "ngang" (để lắt sang hai bên) và phương "dọc" (hướng di chuyển), cũng như phương "pháp tuyến" (hướng vào bề mặt vật liệu).

### 2.1. Vector Tiếp tuyến, Pháp tuyến và Trùng pháp tuyến

Giả sử quỹ đạo hàn được tham số hóa bởi độ dài cung  $s$ , vị trí tâm tại  $s$  là  $P(s)$ . Ta cần xác định bộ ba vector trực giao  $(T, N, B)$  tại mỗi điểm trên quỹ đạo. Tuy nhiên, việc sử dụng khung Frenet-Serret tiêu chuẩn (dựa trên đạo hàm bậc hai của đường cong) thường gặp vấn đề tại các đoạn đường thẳng (nơi độ cong bằng 0, dẫn đến vector pháp tuyến không xác định).<sup>4</sup>

Trong ứng dụng robot hàn, phương pháp tiếp cận mạnh mẽ hơn là sử dụng hướng của mỏ hàn để định nghĩa mặt phẳng dao động.

#### Định nghĩa các Vector:

- Vector Tiếp tuyến ( $T$ ):** Hướng di chuyển của mỏ hàn.

$$T(t) = \frac{P_{i+1} - P_i}{\|P_{i+1} - P_i\|}$$

Trong điều khiển thời gian thực, đây là vector vận tốc tuyến tính chuẩn hóa:

$$T = v / \|v\|.$$

- Vector Hướng Tiếp cận ( $A$ ):** Hướng trục Z của mỏ hàn (Tool Z-axis), hướng từ mặt bích robot đến đầu dây hàn. Đây là thông số đã biết từ động học thuận.
- Vector Dao động ( $W$  - Weave Direction):** Đây là vector quan trọng nhất, xác định phương ngang để mỏ hàn lắt sang trái/phải. Vector này phải vuông góc với hướng di chuyển và nằm trong mặt phẳng mong muốn. Công thức tổng quát nhất để tính  $W$  (tương ứng với vector Trùng pháp tuyến hoặc Binormal trong ngữ cảnh hàn) là tích có hướng của Vector Tiếp tuyến và Vector Hướng Tiếp cận<sup>1</sup>:

$$W = \frac{T \times A}{||T \times A||}$$

Lưu ý: Nếu  $T$  song song với  $A$  (trường hợp hàn chúi xuống thẳng đứng), phép tính này sẽ sinh ra điểm kỳ dị (singularity). Trong thực tế, mỏ hàn luôn có một góc nghiêng (travel angle) hoặc góc làm việc (work angle), nên tích có hướng này thường hợp lệ. Nếu xảy ra kỳ dị, cần sử dụng vector tham chiếu bổ sung (ví dụ: trục Z của hệ tọa độ World).

#### 4. Vector Pháp tuyến Bề mặt Ảo ( $N_{virt}$ ):

$$N_{virt} = W \times T$$

Vector này giúp xác định chiều cao (nếu cần lắc theo phương đứng, ví dụ kiểu Figure-8).

## 2.2. Ma trận Biến đổi Homogeneous

Vị trí thực tế của TCP khi có Weaving ( $P_{out}$ ) được tính bằng vị trí trên đường dẫn gốc ( $P_{ref}$ ) cộng với độ lệch cục bộ ( $d_{local}$ ) đã được quay về hệ tọa độ gốc (Base Frame).<sup>1</sup>

$$P_{out} = P_{ref} + \cdot \begin{bmatrix} Amp \cdot \delta(\phi) \\ 0 \\ 0 \end{bmatrix}$$

Trong đó  $\cdot$  là ma trận quay được tạo bởi các vector cơ sở đã tính ở trên.  $\delta(\phi)$  là hàm dao động chuẩn hóa.

## 3. Kỹ thuật Weaving (Hàn Lắc): Phân tích và Triển khai C++

Weaving là kỹ thuật dao động mỏ hàn ngang qua rãnh hàn để mở rộng bề hàn, tăng khả năng điền đầy và cải thiện sự hòa trộn kim loại.<sup>2</sup> Về mặt toán học, đây là việc tạo ra một tín hiệu tuần hoàn  $\delta(\phi)$  phụ thuộc vào pha  $\phi$ .

### 3.1. Quản lý Pha (Phase) và Tần số

Pha dao động  $\phi$  biến thiên theo thời gian  $t$ . Trong hệ thống rời rạc (digital controller), pha

được cập nhật sau mỗi chu kỳ lấy mẫu  $\Delta t$  (ví dụ: 0.001s).

**Công thức cập nhật pha:**

$$\phi_{k+1} = \phi_k + 2\pi \cdot f \cdot \Delta t$$

Trong đó  $f$  là tần số lấy mẫu (Hz). Pha  $\phi$  thường được chuẩn hóa trong khoảng  $[0, 2\pi)$  hoặc tùy quy ước lập trình.

## 3.2. Các Kiểu Dao động (Patterns) và Công thức Toán học

Dưới đây là chi tiết toán học cho các kiểu Weaving phổ biến được yêu cầu <sup>1</sup>:

### 3.2.1. Sine Wave (Hình Sin)

Đây là dạng cơ bản nhất, tạo ra chuyển động mượt mà, giảm rung động cơ khí cho robot.

- **Ứng dụng:** Hàn phủ (cover pass), yêu cầu bề mặt đẹp.
- **Công thức:**

$$\delta_{\text{sine}}(\phi) = \sin(\phi)$$

$$\text{Offset} = A \cdot \sin(\phi) \cdot W$$

- **Đặc điểm nhiệt:** Nhiệt lượng tập trung nhiều hơn ở biên (nơi vận tốc ngang tiến tới 0) và ít hơn ở giữa (nơi vận tốc ngang cực đại). Điều này tự nhiên giúp chống cháy cạnh (undercut).<sup>1</sup>

### 3.2.2. Triangle Wave (Hình Tam giác)

Chuyển động tuyến tính giữa hai biên.

- **Ứng dụng:** Hàn góc (Fillet), hàn rãnh chữ V, yêu cầu điền đầy đều và ngấu sâu tại góc.<sup>6</sup>
- **Công thức:**

Hàm tam giác chuẩn hóa biên độ  $[-1, 1]$  với chu kỳ  $2\pi$ :

$$\delta_{\text{tri}}(\phi) = \begin{cases} \frac{2}{\pi}\phi & 0 \leq \phi < \frac{\pi}{2} \\ 2 - \frac{2}{\pi}\phi & \frac{\pi}{2} \leq \phi < \frac{3\pi}{2} \\ \frac{2}{\pi}\phi - 4 & \frac{3\pi}{2} \leq \phi < 2\pi \end{cases}$$

- **Vấn đề Gia tốc:** Tại các đỉnh ( $\pm 1$ ), vận tốc đổi chiều tức thời gây ra gia tốc vô hạn về lý thuyết. Trong lập trình C++, cần áp dụng bộ lọc làm mượt (Smoothing/Corner Rounding)

tại các đỉnh này để bảo vệ hộp số robot.<sup>8</sup>

### 3.2.3. Trapezoidal Wave (Hình Thang)

Tương tự hình tam giác nhưng có đoạn nằm ngang tại biên.

- **Ứng dụng:** Tăng cường nung chảy thành bên (sidewall fusion) cho vật liệu dày.
- **Công thức:** Được xác định bởi tham số Dwell Time (xem phần 4). Về mặt hình học, nó bão hòa tại  $\pm 1$ .

### 3.2.4. Circular Wave (Hình Tròn)

Mở hàn vẽ các vòng tròn nhỏ dọc theo đường hàn.

- **Ứng dụng:** Hàn đắp bề mặt (surfacing/cladding), kiểm soát vũng hàn lỏng khi hàn trần (overhead).<sup>2</sup>
- **Công thức:**

Khác với Sine/Triangle chỉ dao động trên trục  $W$ , Circular dao động trên cả trục  $W$  và trục  $T$  (tiếp tuyến).

$$\text{Offset}_W = R \cdot \cos(\phi)$$

$$\text{Offset}_T = R \cdot \sin(\phi)$$

*Lưu ý:* Việc thêm offset vào trục  $T$  làm thay đổi vận tốc di chuyển tức thời của mỏ hàn so với phôi (khi quay ngược chiều di chuyển, mỏ hàn đi chậm lại; khi quay cùng chiều, nó đi nhanh hơn).

### 3.2.5. Figure-8 (Hình số 8)

- **Ứng dụng:** Hàn leo (Vertical Up), vật liệu rất dày. Giúp vũng hàn nguội bớt khi mỏ hàn đảo chiều, ngăn kim loại lỏng chảy xệ.<sup>9</sup>
- **Công thức (Lemniscate of Gerono):**

$$\text{Offset}_W = A \cdot \sin(\phi)$$

$$\text{Offset}_T = A_{\text{long}} \cdot \sin(\phi) \cdot \cos(\phi)$$

Trong đó  $A_{\text{long}}$  là biên độ dọc (thường nhỏ hơn biên độ ngang  $A$ ).

### 3.2.6. L-Type Weaving (Kiểu chữ L)

Đây là kỹ thuật đặc biệt cho hàn góc (Fillet Weld), nơi biên dạng dao động không đối xứng qua

tâm mà bám theo hai cạnh của góc vuông (thành đứng và sàn).<sup>8</sup>

- **Logic:**

Thay vì dao động quanh một điểm trung tâm, L-Type di chuyển từ:

Gốc (Root)  $\rightarrow$  Lên thành đứng (Vertical Wall)  $\rightarrow$  Về Gốc  $\rightarrow$  Ra sàn ngang (Horizontal Floor)  $\rightarrow$  Về Gốc.

- **Vector cơ sở:** Cần xác định hai vector riêng biệt:  $V_{wall}$  (thường là trục Z của Robot hoặc vector dạy trước) và  $V_{floor}$ .

- **Thuật toán:**

- Nếu  $0 \leq \phi < \pi$ : Di chuyển dọc  $V_{wall}$ .
- Nếu  $\pi \leq \phi < 2\pi$ : Di chuyển dọc  $V_{floor}$ .

### 3.3. Giải mã C++ cho Module Weaving

Dưới đây là cấu trúc class và hàm tính toán offset cốt lõi:

C++

```
// Các kiểu Weaving hỗ trợ
enum WeavePattern { SINE, TRIANGLE, TRAPEZOID, CIRCULAR, FIGURE_8, L_TYPE };

struct WeaveParams {
    WeavePattern pattern;
    double amplitude;    // Biên độ (mm)
    double frequency;    // Tần số (Hz)
    double dwellLeft;    // Thời gian dừng trái (s)
    double dwellRight;   // Thời gian dừng phải (s)
    double lTypeWallBias; // Độ lệch cho L-type (mm)
    Vector3 wallVector;   // Vector tường cho L-type
};

class WeldPatternGenerator {
private:
    double currentPhase;
    double cycleTime;    // Thời gian chu kỳ (dt), ví dụ 0.001s

public:
```

```

Vector3 CalculateOffset(Vector3 tangent, Vector3 toolZ, WeaveParams params) {
    // 1. Cập nhật pha dao động
    double phaseStep = 2 * PI * params.frequency * this->cycleTime;
    this->currentPhase += phaseStep;
    if (this->currentPhase >= 2 * PI) this->currentPhase -= 2 * PI;

    // 2. Xác định hệ tọa độ Weaving (Weave Frame)
    Vector3 binormal = CrossProduct(tangent, toolZ).Normalize();
    Vector3 normal = CrossProduct(binormal, tangent).Normalize(); // Vector ảo hướng lên

    double u = 0; // Thành phần dao động ngang (Binormal)
    double v = 0; // Thành phần dao động dọc (Tangent) - dùng cho Circle/Fig8
    double w = 0; // Thành phần pháp tuyến (Normal) - ít dùng

    // 3. Tính toán theo Pattern
    switch (params.pattern) {
        case SINE:
            u = params.amplitude * sin(this->currentPhase);
            break;

        case TRIANGLE:
            // Hàm tam giác tuyến tính hóa
            double p = this->currentPhase; // alias
            if (p < PI/2) u = params.amplitude * (p / (PI/2));
            else if (p < 3*PI/2) u = params.amplitude * (1.0 - (p - PI/2)/(PI/2));
            else u = params.amplitude * (-1.0 + (p - 3*PI/2)/(PI/2));
            break;

        case FIGURE_8:
            u = params.amplitude * sin(this->currentPhase);
            // v lệch pha để tạo hình số 8
            v = (params.amplitude * 0.5) * sin(this->currentPhase) * cos(this->currentPhase);
            break;

        case L_TYPE:
            // Logic riêng cho L-Type: Phụ thuộc vào Wall Vector
            // Trả về offset tính trực tiếp từ Wall/Floor vector, không dùng binormal chuẩn
            return CalculateLTypeOffset(params);
    }

    // 4. Tổng hợp vector offset
    Vector3 finalOffset = (binormal * u) + (tangent * v) + (normal * w);
    return finalOffset;
}

```

```
}  
};
```

## 4. Logic Điều khiển Dwell Time (Thời gian dừng)

Dwell Time là khoảng thời gian mỏ hàn dừng lại (hoặc di chuyển rất chậm) tại hai biên của biên độ lắc. Đây là yếu tố quyết định để tránh khuyết tật "undercut" (cháy cạnh) và đảm bảo độ ngấu thành bên.<sup>1</sup>

### 4.1. Cơ chế Vật lý

Khi hàn lắc, mỏ hàn di chuyển qua trung tâm rãnh hàn với tốc độ cao nhất (đối với Sine) hoặc bằng nhau (Triangle). Tuy nhiên, tại thành bên (sidewall), nhiệt lượng bị tản nhanh vào phôi. Nếu mỏ hàn quay đầu ngay lập tức, nhiệt lượng cung cấp không đủ để làm nóng chảy thành bên, gây ra khuyết tật thiếu ngấu (lack of fusion). Việc "Dwell" giúp cung cấp đủ nhiệt lượng tại biên.

### 4.2. Logic Trạng thái (State Machine) cho Dwell

Trong phần mềm điều khiển, Dwell không chỉ đơn giản là một hàm toán học mà là một máy trạng thái (State Machine) can thiệp vào bộ cập nhật pha.

#### Thuật toán Dwell:

Thay vì chỉ tính  $\delta(\phi)$ , hệ thống cần kiểm tra xem vị trí hiện tại có đạt tới biên độ cực đại hay không.

1. **Trạng thái MOVE:** Tăng pha  $\phi$ . Tính toán vị trí. Nếu  $|\text{offset}| \geq \text{Amplitude}$ , chuyển sang trạng thái DWELL.
2. **Trạng thái DWELL:**
  - Ngưng tăng pha  $\phi$  (giữ nguyên  $\phi$  tại  $\pi/2$  hoặc  $3\pi/2$ ).
  - Kích hoạt bộ đếm thời gian (Timer).
  - Tùy chọn hành vi chuyển động dọc ( $T$ ):
    - **Weave Stop:** Chỉ dừng dao động ngang, robot vẫn tiến về phía trước (tạo ra đường hàn hình thang trên quỹ đạo).
    - **Robot Stop:** Dừng toàn bộ chuyển động robot (TCP đứng yên tuyệt đối). Đây là chế độ phổ biến cho hàn dày.<sup>8</sup>
  - Khi  $\text{Timer} \geq \text{DwellTime}$  cài đặt  $\rightarrow$  Chuyển về trạng thái MOVE, tiếp tục tăng  $\phi$ .



### 4.3. Giả mã Logic Dwell (Tích hợp vào CalculateOffset)

C++

```
if (state == WEAVE_STATE_DWELL_LEFT) {
    dwellTimer += cycleTime;
    if (dwellTimer >= params.dwellLeft) {
        state = WEAVE_STATE_MOVING_RIGHT;
        dwellTimer = 0;
    }
    // Giữ nguyên offset tại biên trái
    u = params.amplitude;
}
else if (state == WEAVE_STATE_DWELL_RIGHT) {
    dwellTimer += cycleTime;
    if (dwellTimer >= params.dwellRight) {
        state = WEAVE_STATE_MOVING_LEFT;
        dwellTimer = 0;
    }
    // Giữ nguyên offset tại biên phải
    u = -params.amplitude;
}
else {
    // Logic di chuyển bình thường...
    // Kiểm tra điều kiện vào Dwell
    if (ReachLeftPeak(currentPhase)) state = WEAVE_STATE_DWELL_LEFT;
    if (ReachRightPeak(currentPhase)) state = WEAVE_STATE_DWELL_RIGHT;
}
```

---

## 5. Kỹ thuật Stitch Welding (Hàn Đính) và Logic Crater Fill

Stitch Welding (hay Intermittent Welding) là kỹ thuật tạo ra các đoạn hàn ngắn ngắt quãng, thường dùng để giảm biến dạng nhiệt trên vật liệu mỏng hoặc gá lắp tạm thời.<sup>14</sup>

### 5.1. Sơ đồ Thời gian (Timing Diagram) và Máy Trạng thái

Stitch Welding yêu cầu phối hợp chặt chẽ giữa chuyển động robot và tín hiệu kích hoạt máy hàn (Arc On/Off).

**Chu trình chuẩn:**

- 1. **Approach:** Robot di chuyển nhanh đến điểm bắt đầu đoạn đính.
- 2. **Arc Start:** Gửi tín hiệu ArcOn. Robot đợi tín hiệu phản hồi ArcEstablished từ nguồn hàn.
- 3. **Weld:** Di chuyển quãng đường StitchLength với vận tốc hàn.
- 4. **Crater Fill (Lấp rãnh):**
  - Khi đến cuối đoạn đính, robot không tắt hồ quang ngay.
  - Robot giảm tốc độ hoặc dừng hẳn.
  - Gửi tín hiệu chuyển nguồn hàn sang chế độ "Crater" (Dòng điện/Điện áp thấp hơn).
  - Duy trì trong thời gian CraterTime.
- 5. **Arc End:** Gửi tín hiệu ArcOff.
- 6. **Burnback:** Đợi dây hàn cháy ngược một chút để không dính vào phôi.
- 7. **Air Move:** Di chuyển nhanh (không hàn) quãng đường Pitch (hoặc Gap) đến đoạn tiếp theo.

**5.2. Logic Crater Fill**

Tại sao cần Crater Fill? Khi hồ quang tắt đột ngột, vũng hàn kim loại lỏng co lại khi nguội, tạo ra một vết lõm (crater) ở cuối đường hàn. Đây là điểm yếu chịu lực và dễ nứt.<sup>14</sup>

**Các chiến thuật điều khiển:**

- **Time-based:** Dừng robot, giảm dòng hàn, giữ trong  $t$  giây.
- **Back-step:** Tại điểm cuối, robot đi lùi lại một khoảng nhỏ (ví dụ 5mm) vào trong vũng hàn rồi mới tắt hồ quang.<sup>18</sup>
- **Ramp-down:** Giảm dần tốc độ cấp dây (Wire Feed Speed) trong khi robot vẫn di chuyển chậm lại.

**5.3. Bảng Dữ liệu Điều khiển Stitch**

Tham số	Đơn vị	Mô tả
Stitch Length	mm	Chiều dài đoạn hàn
Stitch Pitch	mm	Khoảng cách giữa các điểm bắt đầu (hoặc Gap Length)
Crater Time	sec	Thời gian điền đầy rãnh cuối

Crater Volts/Amps	V/A	Thông số hàn cho giai đoạn lấp rãnh
Burnback Time	sec	Thời gian chờ dây cháy ngược

## 6. Chiến lược Multi-pass Offset (Hàn Nhiều Lớp)

Đối với vật liệu dày (thường > 10mm), một đường hàn đơn không thể điền đầy rãnh. Cần thực hiện hàn nhiều lớp (Multi-pass). Thách thức toán học ở đây là tính toán vị trí offset  $(Y, Z)$  cho từng đường hàn (pass) sao cho chúng xếp chồng lên nhau khít khao, không để lại lỗ hổng.<sup>19</sup>

### 6.1. Nguyên lý Hình học và Quy tắc Xếp chồng

Có hai mô hình rãnh hàn chính: V-Groove và Fillet. Nguyên tắc cơ bản là **Quy tắc 50% Overlap**: Đường hàn sau nên đè lên đường hàn trước khoảng 50% bề rộng để đảm bảo bề mặt lớp hàn phẳng.<sup>22</sup>

Diện tích tiết diện ngang của mỗi đường hàn ( $A_{bead}$ ) có thể ước tính xấp xỉ bằng công thức Parabol hoặc hình bán nguyệt dựa trên tốc độ cấp dây ( $WFS$ ) và tốc độ di chuyển ( $TS$ ):

$$A_{bead} \approx \frac{WFS \cdot A_{wire}}{TS} \cdot \eta$$

Trong đó  $A_{wire}$  là diện tích tiết diện dây hàn,  $\eta$  là hiệu suất đắp.

### 6.2. Thuật toán Tính toán Offset cho V-Groove

Giả sử rãnh chữ V có góc  $\alpha$ , khe hở đáy  $G$ . Ta cần xếp các đường hàn vào các lớp (Layers).

**Công thức tính toán:**

- Chiều cao lớp ( $h$ ):** Thường bằng độ dày hiệu dụng của một đường hàn (ví dụ 2-3mm).
- Bề rộng rãnh tại lớp  $n$  ( $W_n$ ):**

$$W_n = G + 2 \cdot (n \cdot h) \cdot \tan\left(\frac{\alpha}{2}\right)$$

### 3. Số đường hàn trong lớp $n$ ( $N_{pass}$ ):

Dựa trên bề rộng đường hàn đơn ( $w$ ) và độ chồng lấn ( $ov \approx 0.5$ ).

$$N_{pass} = \text{ceil} \left( \frac{W_n - w}{w \cdot (1 - ov)} + 1 \right)$$

### 4. Offset ngang ( $Y_{n,i}$ ) cho đường thứ $i$ trong lớp $n$ :

$$Y_{n,i} = -\frac{W_n - w}{2} + (i - 1) \cdot w \cdot (1 - ov)$$

(Tính từ tâm rãnh).

### 5. Offset đứng ( $Z_n$ ):

$$Z_n = n \cdot h$$

## 6.3. Giải mã C++ cho Multi-pass Generator

C++

```
struct GrooveParams {
    double angle; // Góc vát (radian)
    double rootGap; // Khe hở đáy (mm)
    double beadWidth; // Bề rộng mỗi hàn đơn (mm)
    double layerHeight; // Chiều cao mỗi lớp (mm)
    double overlap; // Tỷ lệ chồng lấn (0.0 - 1.0, thường là 0.5)
};

std::vector<Vector3> CalculateMultiPassOffsets(GrooveParams params, int totalLayers) {
    std::vector<Vector3> offsets;

    for (int n = 1; n <= totalLayers; n++) {
        double z = n * params.layerHeight;

        // Tính bề rộng rãnh khả dụng tại độ cao z
        double widthAtLayer = params.rootGap + 2 * z * tan(params.angle / 2.0);

        // Tính số đường hàn cần thiết (dựa trên overlap)
```

```

double effectiveStep = params.beadWidth * (1.0 - params.overlap);
int numPasses = (int)ceil((widthAtLayer - params.beadWidth) / effectiveStep) + 1;
if (numPasses < 1) numPasses = 1; // ít nhất 1 đường

// Tính vị trí bắt đầu (bên trái rãnh)
// Center of bead logic
double totalBeadSpan = params.beadWidth + (numPasses - 1) * effectiveStep;
double startY = -totalBeadSpan / 2.0 + (params.beadWidth / 2.0);

for (int i = 0; i < numPasses; i++) {
    double y = startY + i * effectiveStep;
    offsets.push_back(Vector3(0, y, z));
    // Lưu ý: Đây là offset trong hệ tọa độ Rãnh (Groove Frame).
    // Cần nhân với ma trận quay của đường hàn để ra offset Robot.
}
}
return offsets;
}

```

## 6.4. Các yếu tố ảnh hưởng bậc hai

- **Biến dạng nhiệt:** Khi hàn nhiều lớp, phôi sẽ bị co rút (transverse shrinkage), làm rãnh hẹp lại.<sup>24</sup> Các phần mềm cao cấp thường có tham số bù co ngót, giảm dần  $W_n$  theo số lớp thực tế đo được hoặc dự đoán.
- **Thứ tự hàn:** Để giảm biến dạng, thứ tự hàn trong một lớp thường là so le hoặc từ ngoài vào trong, thay vì chỉ đơn thuần từ trái sang phải.

## 7. Kết luận và Khuyến nghị Triển khai

Việc xây dựng module WeldPatternGenerator cho robot 6 trục là một bài toán phức tạp đòi hỏi sự kết hợp giữa toán học hình học (cho Weaving/Multi-pass) và logic máy trạng thái thời gian thực (cho Dwell/Stitch).

### Các điểm chính cần lưu ý khi lập trình C++:

1. **Tính Modularity:** Tách biệt lớp tính toán offset (PatternGenerator) khỏi lớp nội suy chuyển động (Interpolator). Interpolator chỉ nên nhận đầu vào là vector offset  $\Delta P$  tại mỗi chu kỳ 1ms.
2. **Xử lý ngoại lệ:** Chú ý các điểm kỳ dị khi tính toán vector Binormal (khi mở hàn song song với đường đi). Luôn có vector dự phòng (fallback vector).
3. **Làm mượt (Smoothing):** Các dạng sóng tam giác hoặc hình thang lý thuyết có giạt cục

về gia tốc. Cần áp dụng bộ lọc thông thấp (Low-pass filter) hoặc Spline để làm mượt quỹ đạo trước khi gửi xuống servo.

4. **Tối ưu hóa:** Các hàm lượng giác ( $\sin$ ,  $\cos$ ,  $\text{atan2}$ ) tốn kém CPU. Nên sử dụng bảng tra (Look-up Table) hoặc xấp xỉ chuỗi Taylor nếu vi xử lý điều khiển có tài nguyên hạn chế, tuy nhiên với các controller hiện đại (ARM Cortex-M7 hoặc x86), việc tính toán trực tiếp thường vẫn khả thi.

Bằng cách tuân thủ các công thức và kiến trúc trên, bạn có thể xây dựng một bộ điều khiển hàn robot chuyên nghiệp, có khả năng cạnh tranh với các giải pháp thương mại từ Fanuc hay Yaskawa.

## Works cited

1. An Algorithm for the Welding Torch Weaving Control of Arc Welding Robot - Semantic Scholar, accessed February 1, 2026, <https://pdfs.semanticscholar.org/91d2/5b5cef87617741edce7b7c954f1141293ef7.pdf>
2. Weaving in welding | A comprehensive guide to weave patterns - Smooth Robotics, accessed February 1, 2026, <https://smooth-robotics.com/weaving-in-welding/>
3. Towards a Uniform Welding Quality: A Novel Weaving Welding ..., accessed February 1, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC9181624/>
4. Universal Path-Following of Wheeled Mobile Robots: A Closed-Form Bounded Velocity Solution - PMC - NIH, accessed February 1, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC8624698/>
5. Continuous trajectory planning for welding of complex joints using bezier curve. - TUTDoR, accessed February 1, 2026, <https://tutvital.tut.ac.za/bitstreams/0987ba96-8f46-4f90-97a8-bf53c7d6c4e3/download>
6. Guide to Weaving in Welding: Weave Patterns and Pitch Calculation - Hirebotics Blog, accessed February 1, 2026, <https://blog.hirebotics.com/weave-welding-guide>
7. Welding Weaving Patterns | HALDEN, accessed February 1, 2026, <https://haldencn.com/welding-weaving-patterns/>
8. Weaving Condition Files - Yaskawa Knowledge Center, accessed February 1, 2026, <https://knowledge.motoman.com/hc/en-us/articles/5842915195543-Weaving-Condition-Files>
9. A novel 8-shape trajectory weaving welding control algorithm with auto-adjust welding torch attitude | Request PDF - ResearchGate, accessed February 1, 2026, [https://www.researchgate.net/publication/360232137\\_A\\_novel\\_8-shape\\_trajectory\\_weaving\\_welding\\_control\\_algorithm\\_with\\_auto-adjust\\_welding\\_torch\\_attitude](https://www.researchgate.net/publication/360232137_A_novel_8-shape_trajectory_weaving_welding_control_algorithm_with_auto-adjust_welding_torch_attitude)
10. Schematic diagram of weaving points of circular trajectory. - ResearchGate, accessed February 1, 2026, <https://www.researchgate.net/figure/Schematic-diagram-of-weaving-points-of-c>

[ircular-trajectory\\_fig3\\_353844809](#)

11. Eight Curve -- from Wolfram MathWorld, accessed February 1, 2026, <https://mathworld.wolfram.com/EightCurve.html>
12. Eight Curve (Leminiscate of Gerono) - Statistics How To, accessed February 1, 2026, <https://www.statisticshowto.com/eight-curve/>
13. Weaving FANUC | PDF | Welding | Construction - Scribd, accessed February 1, 2026, <https://www.scribd.com/document/942466275/Weaving-FANUC>
14. Adjusting Crater Fill with ESAB - Beacon Platform - Hirebotics, accessed February 1, 2026, <https://help.hirebotics.com/en/articles/8280465-adjusting-crater-fill-with-esab>
15. MULTI NEXT TACK/STITCH - Koike, accessed February 1, 2026, [https://www.koike.com/content/Manuals/Wel-Handy%20Multi%20Next%20Manual%204\\_19.pdf](https://www.koike.com/content/Manuals/Wel-Handy%20Multi%20Next%20Manual%204_19.pdf)
16. TECH TIP: Programming The Crater Fill Function For Welding Aluminium On SP Series MIG Welders, accessed February 1, 2026, <http://prospotwelding.blogspot.com/2018/06/tech-tip-programming-crater-fill.html>
17. Arc Shots: Close-Up Aluminum MIG Crater Fill Technique - YouTube, accessed February 1, 2026, <https://www.youtube.com/shorts/qlGp9iKzKOs>
18. Filling craters with MIG : r/Welding - Reddit, accessed February 1, 2026, [https://www.reddit.com/r/Welding/comments/1i5dow5/filling\\_craters\\_with\\_mig/](https://www.reddit.com/r/Welding/comments/1i5dow5/filling_craters_with_mig/)
19. Optimization of Welding Parameters Using an Improved Hill-Climbing Algorithm Based on BP Neural Network for Multi-Bead Weld Smoothness Control - NIH, accessed February 1, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC12430673/>
20. COBEM2021-1752 WELDING BEADS OVERLAPPING ALGORITHM DEDICATED TO WAAM - LABSOLDA, accessed February 1, 2026, [https://labsolda.ufsc.br/publicacoes/artigos/2021\\_cobem\\_rocha.pdf](https://labsolda.ufsc.br/publicacoes/artigos/2021_cobem_rocha.pdf)
21. CN111496428B - Multilayer multi-pass welding bead planning method based on straight welding seam contour recognition and welding workstation - Google Patents, accessed February 1, 2026, <https://patents.google.com/patent/CN111496428B/en>
22. Multipass Welding Guide: Techniques, Applications and Best Practices, accessed February 1, 2026, <https://www.arccaptain.com/blogs/article/multipass-welding>
23. I've been welding for about two months now and this is one of my first multi pass fillet welds. Thought? - Reddit, accessed February 1, 2026, [https://www.reddit.com/r/Welding/comments/prp0i8/ive\\_been\\_welding\\_for\\_about\\_two\\_months\\_now\\_and/](https://www.reddit.com/r/Welding/comments/prp0i8/ive_been_welding_for_about_two_months_now_and/)
24. (PDF) A knowledge-based multipass welding distortion estimation method for a multi-robot welding off-line programming and simulation software - ResearchGate, accessed February 1, 2026, [https://www.researchgate.net/publication/347073103\\_A\\_knowledge-based\\_multi\\_pass\\_welding\\_distortion\\_estimation\\_method\\_for\\_a\\_multi-robot\\_welding\\_off-line\\_programming\\_and\\_simulation\\_software](https://www.researchgate.net/publication/347073103_A_knowledge-based_multi_pass_welding_distortion_estimation_method_for_a_multi-robot_welding_off-line_programming_and_simulation_software)
25. 04 - Welding Design., accessed February 1, 2026, <https://www.nrc.gov/docs/ML1215/ML12157A631.pdf>