

# Báo cáo Nghiên cứu Chuyên sâu: Kiến trúc Kỹ thuật và Giải pháp Tích hợp Hệ thống Điều khiển Robot Hàn 6-DOF Thích ứng Thời gian Thực

## 1. Tổng quan Điều hành

Việc phát triển một bộ điều khiển robot hàn 6-DOF thương mại dựa trên nền tảng PC (PC-based Controller) là một thách thức kỹ thuật đa ngành, đòi hỏi sự đồng bộ hóa chính xác giữa thị giác máy tính (Computer Vision), lập kế hoạch quỹ đạo (Trajectory Planning) và điều khiển chuyển động thời gian thực (Real-time Motion Control). Yêu cầu cốt lõi của hệ thống là khả năng "Seam Tracking" (Bám đường hàn) với độ trễ tối thiểu, trong đó robot phải phản ứng tức thời với các sai lệch khe hàn được phát hiện bởi cảm biến laser, đồng thời duy trì sự mượt mà của chuyển động thông qua thuật toán Ruckig để tránh rung động cơ khí (Jerk).

Môi trường hàn hồ quang (Arc Welding) là một trong những môi trường khắc nghiệt nhất đối với các cảm biến quang học do nhiễu quang phổ rộng, cường độ cao từ hồ quang, tia lửa hàn (spatter), khói và nhiễu điện từ (EMI). Để đạt được độ tin cậy cấp thương mại, hệ thống không chỉ dựa vào thuật toán phần mềm mà phải bắt đầu từ việc lựa chọn vật lý quang học chính xác (bước sóng Laser), kiến trúc phần cứng cảm biến (2D Profiler), và chiến lược phân luồng phần mềm trên C# WPF để đảm bảo tính thời gian thực mềm (Soft Real-time) trên Windows.

Báo cáo này cung cấp một phân tích kỹ thuật toàn diện, đi sâu vào từng tầng của hệ thống: từ vật lý của sự tương tác ánh sáng laser xanh (Blue Laser) với bề mặt kim loại nóng chảy, đến việc triển khai toán học của các thuật toán trích xuất đường hàn (RANSAC, Steger), và cuối cùng là kiến trúc tích hợp thư viện Ruckig C++ vào môi trường .NET qua C++/CLI để điều khiển vi điều khiển Teensy 4.1 chạy grblHAL. Đây là lộ trình kỹ thuật chi tiết nhằm xây dựng một hệ thống hàn thông minh, cạnh tranh với các giải pháp công nghiệp cao cấp từ Keyence hay Fanuc nhưng với chi phí tối ưu hóa và khả năng tùy biến cao.

---

## 2. Kiến trúc Hệ thống Điều khiển Thích ứng (Adaptive Control Architecture)

### 2.1 Thách thức về Độ trễ trong Hệ thống PC-based

Trong mô hình điều khiển PC-based, máy tính đảm nhiệm vai trò của một "Soft-PLC". Tuy nhiên, Windows không phải là hệ điều hành thời gian thực (RTOS). Các tác vụ như Garbage

Collection (GC) của .NET hay việc lập lịch luồng (Thread Scheduling) của Windows có thể gây ra độ trễ (jitter) từ 10-50ms. Đối với hàn robot, độ trễ tổng thể từ lúc cảm biến nhìn thấy sai lệch đến lúc đầu mỏ hàn điều chỉnh vị trí là yếu tố quyết định chất lượng mối hàn.

### Chu trình điều khiển khép kín (Closed-loop Cycle):

1. **Thu thập dữ liệu (Acquisition):** Cảm biến Laser chụp ảnh biên dạng đường hàn (Profile).
2. **Xử lý ảnh (Processing):** Trích xuất tọa độ tâm khe hàn  $(y_{seam}, z_{seam})$  và tính toán vector sai lệch  $\Delta E$ .
3. **Biến đổi tọa độ (Transformation):** Chuyển đổi  $\Delta E$  từ Hệ tọa độ Cảm biến  $\{S\}$  sang Hệ tọa độ Cơ sở Robot  $\{B\}$  thông qua ma trận Hand-Eye Calibration.
4. **Tạo quỹ đạo (Trajectory Generation - Ruckig):** Tính toán trạng thái mục tiêu kế tiếp  $(P_{next}, V_{next}, A_{next})$  dựa trên sai lệch và giới hạn động học.
5. **Truyền thông (Communication):** Gửi lệnh vị trí tới Teensy 4.1.
6. **Thực thi (Execution):** grblHAL phát xung bước (Step/Dir) tới các Servo Driver.

Để đảm bảo robot điều chỉnh "ngay lập tức", chu trình này phải hoạt động ổn định ở tần số tối thiểu 50Hz-100Hz (tức là 10-20ms/chu kỳ). Việc tích hợp Ruckig đóng vai trò là bộ đệm thông minh, giúp làm mịn các lệnh điều khiển bị gián đoạn do jitter của Windows, đảm bảo Teensy luôn nhận được một luồng quỹ đạo khả thi về mặt vật lý.<sup>1</sup>

## 2.2 Vai trò cốt lõi của Ruckig OTG

Ruckig là thư viện tạo quỹ đạo trực tuyến (Online Trajectory Generation - OTG) loại V, cho phép tính toán quỹ đạo tối ưu về thời gian trong khi bị giới hạn bởi Vận tốc, Gia tốc và Độ giật (Jerk).<sup>1</sup>

Trong ngữ cảnh Seam Tracking:

- **Vấn đề:** Các thuật toán điều khiển PID truyền thống thường cộng trực tiếp sai số vào vị trí hiện tại, gây ra sự thay đổi đột ngột về vận tốc, dẫn đến robot bị rung (jerky motion) và đường hàn bị gợn sóng.
- **Giải pháp Ruckig:** Thay vì điều khiển vị trí trực tiếp, hệ thống PC cung cấp cho Ruckig một "Trạng thái Mục tiêu Mới" (New Target State) bao gồm vị trí đã bù sai lệch. Ruckig sẽ tính toán quỹ đạo để chuyển từ trạng thái chuyển động hiện tại sang trạng thái mới một cách mượt mà nhất có thể trong vòng một chu kỳ điều khiển (ví dụ 10ms), đảm bảo các giới hạn cơ khí không bị vi phạm.<sup>4</sup>

---

## 3. Công nghệ Cảm biến Laser: Vật lý và Lựa chọn Phần

## cứng

Việc lựa chọn cảm biến laser quyết định 80% độ ổn định của hệ thống Seam Tracking. Trong môi trường hàn, "kẻ thù" chính là ánh sáng hồ quang (Arc Light).

### 3.1 Vật lý của Nhiễu Hồ quang và Bước sóng Laser

Hồ quang hàn (MIG/MAG/TIG) phát ra bức xạ quang phổ rộng, tuân theo định luật Planck về bức xạ vật đen nhưng có các đỉnh phát xạ cực mạnh đặc trưng cho khí bảo vệ và hơi kim loại.

#### 3.1.1 Tại sao Laser Đỏ (635-660nm) thất bại?

Hầu hết các cảm biến giá rẻ sử dụng laser đỏ. Tuy nhiên, hồ quang hàn và vũng hàn nóng chảy (molten pool) phát ra bức xạ nhiệt rất mạnh ở vùng quang phổ đỏ và hồng ngoại (IR).<sup>6</sup> Điều này dẫn đến tỷ số Tín hiệu trên Nhiễu (SNR) thấp. Cảm biến sẽ bị "mù" do cường độ ánh sáng nền từ hồ quang lớn hơn cường độ tia laser phản xạ, khiến thuật toán không thể tách biên dạng.

#### 3.1.2 Ưu việt của Laser Xanh (405-450nm)

Công nghệ Blue Laser là giải pháp tiêu chuẩn cho hàn robot hiện đại vì hai lý do vật lý cơ bản:

1. **Đặc tính Bức xạ Nhiệt:** Ở nhiệt độ nóng chảy của thép (khoảng 1500°C - 2000°C), bức xạ nhiệt tập trung chủ yếu ở vùng hồng ngoại và giảm dần về phía cực tím. Cường độ bức xạ nền ở bước sóng 450nm (xanh dương) thấp hơn đáng kể so với 650nm (đỏ), giúp tia laser "nổi bật" hơn trên nền nhiễu.<sup>8</sup>
2. **Hệ số Hấp thụ và Phản xạ:** Các kim loại, đặc biệt là kim loại có độ bóng cao như Nhôm hoặc Thép không gỉ, có hệ số hấp thụ ánh sáng xanh cao hơn ánh sáng đỏ. Điều này hạn chế hiện tượng tán xạ bề mặt (sub-surface scattering) và tạo ra đường laser sắc nét hơn, giúp thuật toán trích xuất tâm đường (centerline extraction) đạt độ chính xác cao hơn.<sup>8</sup>

**Kết luận kỹ thuật:** Để hệ thống thương mại hoạt động ổn định, việc sử dụng **Laser Xanh (Blue Laser)** kết hợp với **Kính lọc thông dải hẹp (Narrow Bandpass Filter)** là yêu cầu bắt buộc.<sup>10</sup>

### 3.2 Phân tích Phân khúc Phần cứng (1D vs 2D)

- **Cảm biến 1D (Laser Displacement):** Chỉ đo khoảng cách tại 1 điểm. Không thể xác định biên dạng (profile) của khe hàn chữ V hay Lap joint. Hoàn toàn không phù hợp cho Seam Tracking dẫn hướng ngang.<sup>12</sup>
- **Cảm biến 2D (Laser Profiler):** Chiếu một tia laser dạng vạch (line) và thu về một mảng tọa độ (X, Z) mô tả mặt cắt ngang của khe hàn. Đây là tiêu chuẩn bắt buộc cho Seam Tracking.<sup>12</sup>

### 3.3 Đánh giá Thị trường và Khuyến nghị Phần cứng

Để xây dựng bộ điều khiển thương mại, chúng ta cần cân bằng giữa Hiệu năng, Chi phí và Khả năng tích hợp (SDK).

### 3.3.1 Phân khúc High-End (Keyence, SICK, SmartRay)

Đây là các giải pháp "Turnkey" (chìa khóa trao tay) với độ tin cậy tuyệt đối nhưng chi phí rất cao và khả năng tùy biến sâu hạn chế.

- **Keyence LJ-X8000:** Dòng sản phẩm đầu bảng với khả năng lấy mẫu 3200 profile/giây. Sử dụng cảm biến CMOS tùy biến để xử lý dải động sáng (HDR) cực rộng.
  - *Ưu điểm:* Độ chính xác micron, SDK C# rất ổn định.
  - *Nhược điểm:* Giá thành quá cao (\$10,000+), giao thức đóng, khó can thiệp vào luồng xử lý ảnh thô.<sup>12</sup>
- **SICK Ranger3:** Tốc độ quét cực cao (46kHz). Cung cấp GenIStream API hỗ trợ C# tốt. Tuy nhiên, việc thiết lập phức tạp.<sup>16</sup>

### 3.3.2 Phân khúc Mid-Range (Lựa chọn tối ưu cho OEM)

Phân khúc này cung cấp phần cứng công nghiệp chuẩn (IP67) với giá thành hợp lý và hỗ trợ các giao thức mở (GigE Vision), phù hợp nhất cho việc phát triển bộ điều khiển riêng.

- **Hikrobot (Dòng MV-DL/DP):** Các cảm biến Laser Profiler 3D của Hikrobot sử dụng laser xanh 405nm, hỗ trợ trích xuất độ chính xác 0.05 pixel.
  - *SDK:* Cung cấp VisionMaster và C# SDK đầy đủ, cho phép truy cập cả dữ liệu đám mây điểm (Point Cloud) và dữ liệu ảnh thô.<sup>18</sup>
  - *Lợi thế:* Chi phí cạnh tranh (\$2,000 - \$4,000), hệ sinh thái hỗ trợ lập trình viên mạnh.
- **Mech-Mind (Mech-Eye LNX):** Dòng LNX-7500/8000 chuyên dụng cho kiểm tra và hàn.
  - *Tích hợp:* Cung cấp mã nguồn mẫu C# (GitHub mecheye\_csharp\_samples) rất chi tiết, giúp giảm thời gian phát triển driver.<sup>20</sup>
  - *Tính năng:* Tốc độ quét 4kHz, độ phân giải 4K, thiết kế tối ưu cho phản xạ kim loại.<sup>22</sup>
- **Riftek (RF627Weld):** Nhà sản xuất chuyên về cảm biến đo lường laser. Dòng RF627Weld được thiết kế riêng cho robot hàn với các thuật toán bám đường hàn tích hợp sẵn bên trong cảm biến (Smart Camera).
  - *SDK:* Mã nguồn mở hoàn toàn RF62X-SDK trên GitHub (C++, C# wrapper), là tài liệu tham khảo tuyệt vời cho kiến trúc driver.<sup>23</sup>

### 3.3.3 Phân khúc DIY (Tự xây dựng)

Sử dụng Camera công nghiệp + Module Laser rời.

- **Cấu hình:** Camera Global Shutter (như Basler ace 2 hoặc Daheng Mercury) + Ống kính có gắn Filter quang học 450nm (OD4) + Module Laser Xanh công suất cao (100mW+).
- **Thách thức:**
  - Phải tự thực hiện cân chỉnh hệ thống (Calibration) theo mô hình Scheimpflug để đảm bảo nét trên toàn bộ dải đo.<sup>24</sup>
  - Vấn đề cơ khí: Sự giãn nở nhiệt của giá đỡ in 3D hoặc nhôm phay CNC có thể làm sai

lệch phép đo tam giác lượng giác (Triangulation).<sup>26</sup>

- **Khuyến nghị:** Chỉ nên dùng cho nghiên cứu hoặc Prototype. Đối với sản phẩm thương mại, rủi ro về độ ổn định cơ khí và quang học là quá lớn so với chi phí tiết kiệm được.

**Kết luận lựa chọn:** Đề xuất sử dụng **Hikrobot MV-DL series** hoặc **Mech-Mind LNX**. Đây là giải pháp cân bằng nhất: Cung cấp phần cứng quang học chuẩn công nghiệp (Blue Laser, IP67) và SDK mở để tích hợp sâu vào ứng dụng C# WPF của bạn mà không cần lo lắng về việc thiết kế quang học.

---

## 4. Thuật toán Xử lý Ảnh và Trích xuất Đường hàn

Để đạt được tính năng Seam Tracking thời gian thực, phần mềm trên PC phải xử lý dữ liệu từ cảm biến và trích xuất tọa độ điểm hàn (Feature Point) chính xác.

### 4.1 Kỹ thuật Chống nhiễu Quang học và Số hóa

Trước khi áp dụng thuật toán, dữ liệu hình ảnh phải được làm sạch khỏi nhiễu hồ quang và tia lửa.

- **Lọc quang học (Bandpass Filter):** Sử dụng kính lọc giao thoa (Interference Filter) với bước sóng trung tâm 450nm và băng thông hẹp ( $\pm 10\text{nm}$ ). Kính lọc này chặn  $>99\%$  ánh sáng hồ quang, chỉ cho phép ánh sáng laser đi qua.<sup>11</sup>
- **Kỹ thuật ROI Động (Dynamic ROI):** Không xử lý toàn bộ khung hình. Dựa vào vị trí đường hàn ở frame trước đó ( $t-1$ ), thuật toán dự đoán vị trí ở frame hiện tại ( $t$ ) và chỉ xử lý một vùng nhỏ xung quanh đó. Điều này loại bỏ hoàn toàn nhiễu hồ quang thường xuất hiện ở phía sau đường laser và tăng tốc độ xử lý.<sup>28</sup>
- **Xử lý hình thái học (Morphological Operations):** Sử dụng phép co (Erosion) để loại bỏ các điểm nhiễu nhỏ (spatter) và phép giãn (Dilation) để khôi phục độ liên mạch của đường laser.<sup>29</sup>

### 4.2 Thuật toán Trích xuất Tâm đường Laser (Centerline Extraction)

Độ chính xác của phép đo phụ thuộc vào việc tìm tâm của đường laser với độ phân giải dưới điểm ảnh (sub-pixel).

1. **Phương pháp Trọng tâm (Center of Gravity - CoG):** Tính trung bình trọng số cường độ sáng theo cột. Nhanh nhưng dễ bị sai lệch nếu đường laser bị lóa hoặc phản xạ không đều.<sup>30</sup>
2. **Thuật toán Steger:** Đây là tiêu chuẩn vàng cho độ chính xác. Thuật toán sử dụng ma trận Hessian (đạo hàm bậc hai của cường độ sáng) để tìm điểm cực trị cục bộ theo hướng pháp tuyến của đường cong.
  - *Nguyên lý:* Tại tâm đường laser, đạo hàm bậc nhất bằng 0 và đạo hàm bậc hai có giá trị cực tiểu (độ cong lớn nhất).

- *Triển khai*: Steger rất nặng về tính toán. Cần triển khai bằng C++ (sử dụng OpenCV hoặc thư viện riêng) và bọc (wrap) lại cho C# để đảm bảo tốc độ thời gian thực.<sup>32</sup>

### 4.3 Nhận dạng Đặc trưng Khe hàn (Feature Extraction)

Sau khi có profile 2D (tập hợp các điểm  $Y, Z$ ), hệ thống cần xác định điểm gốc hàn.

#### 4.3.1 Khe hàn chữ V (V-Groove / Butt Joint)

Biên dạng bao gồm hai mặt phẳng tấm và hai mặt vát.

- **Phương pháp RANSAC (Random Sample Consensus)**: Đây là giải pháp tối ưu để loại bỏ nhiễu. RANSAC hoạt động bằng cách chọn ngẫu nhiên một tập hợp điểm con để khớp (fit) một đường thẳng, sau đó đếm số lượng điểm nằm gần đường thẳng đó (inliers).
  - *Quy trình*:
    1. Chia profile thành 2 phần trái/phải.
    2. Dùng RANSAC khớp đường thẳng cho sườn dốc trái ( $L_1$ ) và sườn dốc phải ( $L_2$ ). Các điểm nhiễu do tia lửa hàn (outliers) sẽ tự động bị loại bỏ vì chúng không nằm trên đường thẳng hình học.
    3. Giao điểm của  $L_1$  và  $L_2$  chính là tâm khe hàn.<sup>34</sup>
- **Mã nguồn tham khảo**: GitHub ikh-innovation/roboweldar-weld-seam-detection sử dụng các kỹ thuật tương tự để đề xuất quỹ đạo hàn từ dữ liệu 3D.<sup>36</sup>

#### 4.3.2 Mối hàn Chồng (Lap Joint)

Biên dạng có dạng bậc thang.

- **Phương pháp**: Tìm điểm thay đổi gradient đột ngột. Điểm đặc trưng thường là góc vuông của tấm trên. Cần áp dụng một vector bù (offset) để đưa mỏ hàn vào đúng vị trí góc chân mối hàn.<sup>38</sup>

### 4.4 Tiếp cận Deep Learning (Học sâu)

Đối với các trường hợp nhiễu cực lớn mà thuật toán hình học thất bại, Deep Learning là giải pháp thay thế.

- **Mô hình**: Sử dụng mạng **U-Net** (phân đoạn ảnh) hoặc **YOLOv8-pose** (phát hiện điểm keypoint).
  - **Ưu điểm**: Có khả năng "học" được hình dạng đường hàn ngay cả khi bị che khuất bởi khói hoặc nhiễu mạnh.<sup>39</sup>
  - **Triển khai**: Sử dụng **ONNX Runtime** trong C# để chạy mô hình đã huấn luyện (PyTorch). Cần GPU rời (NVIDIA RTX 3060 trở lên) để đạt tốc độ suy luận < 10ms.<sup>41</sup>
-

## 5. Tích hợp Ruckig và Điều khiển Thời gian Thực

Đây là phần quan trọng nhất để đáp ứng yêu cầu "điều chỉnh quỹ đạo ngay lập tức".

### 5.1 Kiến trúc Phần mềm Đa luồng (Multithreaded Architecture)

Hệ thống PC (C# WPF) cần được thiết kế với mô hình phân luồng nghiêm ngặt:

1. **Luồng UI (WPF):** Hiển thị hình ảnh, profile, thông số. Độ ưu tiên thấp.
2. **Luồng Vision (C++ Backend):** Thu thập ảnh từ GigE Camera, chạy thuật toán Steger/RANSAC. Tần số > 50Hz.
3. **Luồng Điều khiển (Real-time Control Loop):** Chạy Ruckig và giao tiếp với Teensy. Độ ưu tiên ThreadPriority.Highest.

### 5.2 Tích hợp Ruckig (C++) vào C#

Ruckig được viết bằng C++ và không có thư viện.NET chính thức. Để sử dụng hiệu năng cao nhất, không nên dùng P/Invoke đơn thuần.

- **Giải pháp C++/CLI Wrapper:** Tạo một dự án trung gian bằng ngôn ngữ C++/CLI (Managed C++).
  - Dự án này liên kết tĩnh (static link) với thư viện Ruckig gốc (ruckig.lib).
  - Tạo một lớp.NET (ref class RuckigNet) bọc lấy đối tượng C++ ruckig::Ruckig.
  - Dữ liệu (mảng vị trí/vận tốc) được truyền trực tiếp qua con trỏ bộ nhớ (pinned memory) giữa C# và C++, giảm thiểu chi phí copy dữ liệu (marshalling overhead) xuống mức micro giây.<sup>42</sup>

### 5.3 Chiến lược Điều khiển Vòng lặp (Control Loop Strategy)

Chu kỳ điều khiển đề xuất là 10ms (100Hz).

1. **Nhận Phản hồi:** Teensy gửi vị trí hiện tại  $(P_{curr}, V_{curr})$  về PC.
2. **Tính sai lệch:** Vision Thread cung cấp vector sai lệch mới nhất  $\Delta E$ .
3. **Cập nhật Ruckig:**
  - Thay vì cộng dồn sai số vào vị trí, ta cập nhật **Trạng thái Mục tiêu** (Target State) của Ruckig.
  - Gọi hàm ruckig.update() với đầu vào là trạng thái hiện tại và mục tiêu mới.
  - Ruckig sẽ tính toán quỹ đạo chuyển tiếp tối ưu, đảm bảo gia tốc và độ giật nằm trong giới hạn cho phép, giúp robot chuyển hướng mượt mà "như lụa".<sup>1</sup>
4. **Gửi Lệnh:** Kết quả vị trí tiếp theo  $(P_{next})$  được gửi xuống Teensy.

### 5.4 Giao thức PC <-> Teensy (grblHAL)

Giao tiếp qua G-code dạng văn bản (ASCII) quá chậm cho ứng dụng thời gian thực này.

- **Giao thức nhị phân:** Sử dụng giao thức tùy biến qua **USB RawHID** (tốc độ 480Mbps) hoặc **Ethernet UDP**.
- **Sửa đổi grblHAL:** Cần viết một Plugin cho grblHAL để nhận gói tin vị trí thời gian thực (Real-time Overlay) và cộng trực tiếp vào bộ tạo bước (Stepper Generator), bỏ qua bộ đệm lập kế hoạch (Planner Buffer) thông thường để đạt phản ứng tức thời.<sup>1</sup>

## 6. Tài nguyên Mã nguồn Mở (Open Source)

Dưới đây là các kho lưu trữ GitHub được chọn lọc kỹ lưỡng để hỗ trợ phát triển:

### 6.1 Thị giác máy tính & Trích xuất đường laser

- **kam3k/laser\_line\_extraction (C++):** Triển khai thuật toán "Split-and-Merge" và khớp đường thẳng. Rất hữu ích để tham khảo logic xử lý profile.<sup>45</sup>
- **wakkaliu/Steger-Centerline (C++):** Mã nguồn triển khai thuật toán Steger để trích xuất tâm đường laser với độ chính xác sub-pixel. Cần bọc lại để dùng trong C#.<sup>32</sup>
- **ikh-innovation/roboweldar-weld-seam-detection:** Dự án hoàn chỉnh về phát hiện đường hàn sử dụng Deep Learning và xử lý đám mây điểm. Tài liệu tham khảo quý giá cho logic phát hiện khe hàn phức tạp.<sup>36</sup>

### 6.2 SDK Cảm biến & Điều khiển

- **RIFTEK-LLC/RF62X-SDK (C#/C++):** Ví dụ mẫu mực về cách viết driver cho cảm biến laser. Cung cấp cả wrapper C# hoàn chỉnh, có thể học hỏi kiến trúc để áp dụng cho các cảm biến khác.<sup>23</sup>
- **MechMindRobotics/mecheye\_csharp\_samples:** Các ví dụ kết nối và thu thập dữ liệu từ cảm biến Mech-Eye LNX bằng C#.<sup>21</sup>

### 6.3 Thư viện Quỹ đạo

- **pantor/ruckig (C++):** Thư viện gốc. Bắt buộc phải tải về và biên dịch. Chú ý phiên bản Community (miễn phí) so với Pro.<sup>1</sup>
- **anderssandstrom/ruckig-ess:** Một nỗ lực tạo wrapper cho Ruckig, có thể dùng để tham khảo cách viết C++/CLI.<sup>46</sup>

## 7. Lộ trình Triển khai và Kết luận

### 7.1 Bảng so sánh và Đề xuất Phần cứng

Đặc tính	High-End (Keyence/SICK)	Mid-Range (Hikrobot/Mech-	DIY (Camera + Laser)
----------	----------------------------	------------------------------	-------------------------



		Mind)	
Giá thành	Rất cao (\$5k - \$15k)	Hợp lý (\$2k - \$4k)	Thấp (\$500 - \$1k)
Chống nhiễu Hồ quang	Xuất sắc (Phần cứng chuyên dụng)	Rất tốt (Blue Laser + Filter)	Kém/Trung bình (Tùy thiết kế)
Tích hợp SDK	Tốt nhưng đóng	Mở, chuẩn GigE Vision/GenICam	Phải tự viết toàn bộ
Độ phức tạp Dev	Thấp	Trung bình	Rất cao (Rủi ro lớn)

**Đề xuất:** Chọn **Hikrobot MV-DL2025-04H** (Blue Laser). Đây là sự cân bằng tốt nhất giữa hiệu năng công nghiệp và khả năng lập trình tùy biến trên C#.

## 7.2 Tổng kết

Để xây dựng thành công bộ điều khiển robot hàn 6-DOF thương mại này, bạn cần tiếp cận theo mô hình lai (Hybrid Architecture):

- Phần cứng:** Kiên quyết sử dụng **Blue Laser** và kính lọc quang học dải hẹp để giải quyết vấn đề vật lý tại nguồn.
- Thuật toán:** Kết hợp sức mạnh của **C++ (Backend xử lý ảnh/Ruckig)** với sự linh hoạt của **C# WPF (Frontend/Logic)** thông qua lớp C++/CLI. Sử dụng RANSAC cho độ bền vững trước nhiễu spatter.
- Điều khiển:** Tận dụng **Ruckig** để biến các tín hiệu hiệu chỉnh giật cục từ cảm biến thành các chuyển động mượt mà, đảm bảo chất lượng mối hàn.

Hệ thống đề xuất này không chỉ khả thi về mặt kỹ thuật mà còn có tiềm năng cạnh tranh mạnh mẽ về chi phí so với các giải pháp ngoại nhập, miễn là sự chú trọng đúng mức được đặt vào việc tối ưu hóa độ trễ truyền thông giữa PC và Teensy.

### Works cited

- pantor/ruckig: Motion Generation for Robots and Machines. Real-time. Jerk-constrained. Time-optimal. - GitHub, accessed February 1, 2026, <https://github.com/pantor/ruckig>
- RSS 2021, Spotlight Talk 24: Jerk-limited Real-time Trajectory Generation with Arbitrary Target... - YouTube, accessed February 1, 2026, [https://www.youtube.com/watch?v=b6y\\_3JvGFJo](https://www.youtube.com/watch?v=b6y_3JvGFJo)
- Ruckig - Motion Generation for Robots and Machines, accessed February 1, 2026, <https://ruckig.com/>

4. Example 03: Waypoints Trajectory - Ruckig, accessed February 1, 2026, [https://docs.ruckig.com/md\\_pages\\_2example\\_03.html](https://docs.ruckig.com/md_pages_2example_03.html)
5. Jerk-limited Real-time Trajectory Generation with Arbitrary Target States - arXiv, accessed February 1, 2026, <https://arxiv.org/pdf/2105.04830>
6. Laser Wavelength: Wavelength Factors, How Does It Vary? - Xometry, accessed February 1, 2026, <https://www.xometry.com/resources/sheet/laser-wavelength/>
7. Signal processing: spectrum of the interference signal with (red) and without (blue) background - ResearchGate, accessed February 1, 2026, [https://www.researchgate.net/figure/Signal-processing-spectrum-of-the-interference-signal-with-red-and-without-blue\\_fig3\\_258521623](https://www.researchgate.net/figure/Signal-processing-spectrum-of-the-interference-signal-with-red-and-without-blue_fig3_258521623)
8. Blue Laser vs Red Laser: Which One Is Best for You - Snapmaker, accessed February 1, 2026, <https://www.snapmaker.com/blog/blue-laser-vs-red-laser/>
9. LASER SEAM TRACKING SYSTEM FOR WELDING AUTOMATION RF627Weld Series, accessed February 1, 2026, [https://riftek.com/upload/iblock/e95/Laser\\_Seam\\_Tracking\\_System\\_for\\_Welding\\_Automation\\_eng.pdf](https://riftek.com/upload/iblock/e95/Laser_Seam_Tracking_System_for_Welding_Automation_eng.pdf)
10. Hard-Coated Bandpass Filters for Machine Vision Lenses - Thorlabs, accessed February 1, 2026, <https://www.thorlabs.com/hard-coated-bandpass-filters-for-machine-vision-lenses>
11. 450nm CWL, 12.5mm Dia. Hard Coated OD 4.0 25nm Bandpass Filter - Edmund Optics, accessed February 1, 2026, <https://www.edmundoptics.com/p/450nm-cwl-125mm-dia-hard-coated-od-4-25nm-bandpass-filter/29045/>
12. How to Select the Right Laser Seam Tracking Solution | KEYENCE America, accessed February 1, 2026, <https://www.keyence.com/products/measure/resources/measurement-sensors-resources/how-to-select-the-right-laser-seam-tracking-solution.jsp>
13. Clearing the Air: Seam Tracking and Seam Finding - Augmentus, accessed February 1, 2026, <https://www.augmentus.tech/clearing-the-air-seam-tracking-and-seam-finding/>
14. Weld Joint Finding Methods Explained - ABIBLOG, accessed February 1, 2026, <https://blog.binzel-abicor.com/usa/weld-joint-finding-methods-explained>
15. Welding Seam Inspection | Automotive Industry | KEYENCE America, accessed February 1, 2026, <https://www.keyence.com/products/measure/industries/automotive/welding-seam-inspection.jsp>
16. SICK Stream Software - Engineering Tools, accessed February 1, 2026, <https://www.sick.com/my/en/catalog/products/digital-services-and-software/engineering-tools/sick-stream-software/c/g609542?tab=overview>
17. OPERATINGINSTRUCTIONS | Ranger3 | 3D machine vision - SICK, accessed February 1, 2026, [https://www.sick.com/media/docs/3/63/063/operating\\_instructions\\_ranger3\\_3d\\_vision\\_en\\_im0080063.pdf](https://www.sick.com/media/docs/3/63/063/operating_instructions_ranger3_3d_vision_en_im0080063.pdf)
18. 3D Laser Profile Sensor - Hikrobot - Machine Vision Products, accessed February

- 1, 2026,  
<https://www.hikrobotics.com/en/machinevision/visionproduct/?typeld=99&id=157>
19. Machine Vision - Service - Download Center - Hikrobot, accessed February 1, 2026, <https://www.hikrobotics.com/en/machinevision/service/download/>
  20. High-Resolution 3D Laser Profilers | Advanced 3D Scanner | Mech-Mind Robotics, accessed February 1, 2026, <https://www.mech-mind.com/products/mech-eye-3d-laser-profilers.html>
  21. MechMindRobotics/mecheye\_csharp\_samples: C# samples for Mech-Eye camera - GitHub, accessed February 1, 2026, [https://github.com/MechMindRobotics/mecheye\\_csharp\\_samples](https://github.com/MechMindRobotics/mecheye_csharp_samples)
  22. Mech-Eye LNX 3D Laser Profiler | 4K Resolution for High-Accuracy Scanning, accessed February 1, 2026, <https://www.mech-mind.com/news/mech-eye-lnx-4k-resolution-3d-laser-profiler.html>
  23. RIFTEK-LLC/RF62X-SDK - GitHub, accessed February 1, 2026, <https://github.com/RIFTEK-LLC/RF62X-SDK>
  24. A Camera Intrinsic Matrix-Free Calibration Method for Laser Triangulation Sensor - PMC, accessed February 1, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC7830204/>
  25. Laser triangulation measurement system with Scheimpflug calibration based on the Monte Carlo optimization strategy - Optica Publishing Group, accessed February 1, 2026, <https://opg.optica.org/abstract.cfm?uri=oe-30-14-25290>
  26. DIY Laser line experiments using a hard drive, webcam, and open-source software., accessed February 1, 2026, [https://www.youtube.com/watch?v=9\\_YwvB7W1kM](https://www.youtube.com/watch?v=9_YwvB7W1kM)
  27. Low-Cost Automation: An Open Source Laser-Triangulation Sensor based on ROS 2, accessed February 1, 2026, [https://www.researchgate.net/publication/369039705\\_Low-Cost\\_Automation\\_An\\_Open\\_Source\\_Laser-Triangulation\\_Sensor\\_based\\_on\\_ROS\\_2](https://www.researchgate.net/publication/369039705_Low-Cost_Automation_An_Open_Source_Laser-Triangulation_Sensor_based_on_ROS_2)
  28. Real-Time Seam Extraction Using Laser Vision Sensing: Hybrid Approach with Dynamic ROI and Optimized RANSAC - PMC - NIH, accessed February 1, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC12157130/>
  29. A Study on Development of Seam Tracking Algorithm in Robotic GMA Welding Process - Semantic Scholar, accessed February 1, 2026, <https://pdfs.semanticscholar.org/a197/7aa2d889468a8d1b16d5c56325b3375f96da.pdf>
  30. Sub-Pixel Extraction of Laser Stripe Center Using an Improved Gray-Gravity Method - NIH, accessed February 1, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC5422175/>
  31. Adaptive Bidirectional Gray-Scale Center of Gravity Extraction Algorithm of Laser Stripes, accessed February 1, 2026, <https://www.mdpi.com/1424-8220/22/24/9567>
  32. wakkaliu/Steger-Centerline - GitHub, accessed February 1, 2026, <https://github.com/wakkaliu/Steger-Centerline>
  33. A robust method to extract a laser stripe centre based on grey level moment - ResearchGate, accessed February 1, 2026,

- [https://www.researchgate.net/publication/272391819\\_A\\_robust\\_method\\_to\\_extract\\_a\\_laser\\_stripe\\_centre\\_based\\_on\\_grey\\_level\\_moment](https://www.researchgate.net/publication/272391819_A_robust_method_to_extract_a_laser_stripe_centre_based_on_grey_level_moment)
34. Line fitting with RANSAC - University of Illinois, accessed February 1, 2026, <http://luthuli.cs.uiuc.edu/~daf/Courses/CV2026/Slides/Feb26/PDF/ransac.pdf>
  35. A Novel Seam Tracking Technique with a Four-Step Method and Experimental Investigation of Robotic Welding Oriented to Complex Welding Seam - MDPI, accessed February 1, 2026, <https://www.mdpi.com/1424-8220/21/9/3067>
  36. thillRobot/seam\_detection: ROS package for weld seam detection using pointcloud data - GitHub, accessed February 1, 2026, [https://github.com/thillRobot/seam\\_detection](https://github.com/thillRobot/seam_detection)
  37. ikh-innovation/roboweldar-weld-seam-detection - GitHub, accessed February 1, 2026, <https://github.com/ikh-innovation/roboweldar-weld-seam-detection>
  38. A Deep Semantic Segmentation Approach to Accurately Detect Seam Gap in Fixtured Workpiece Laser Welding - MDPI, accessed February 1, 2026, <https://www.mdpi.com/2504-4494/9/3/69>
  39. A lightweight deep learning method for real-time weld feature extraction under strong noise, accessed February 1, 2026, [https://www.researchgate.net/publication/382826562\\_A\\_lightweight\\_deep\\_learning\\_method\\_for\\_real-time\\_weld\\_feature\\_extraction\\_under\\_strong\\_noise](https://www.researchgate.net/publication/382826562_A_lightweight_deep_learning_method_for_real-time_weld_feature_extraction_under_strong_noise)
  40. WeldLight: A Lightweight Weld Classification and Feature Point Extraction Model for Weld Seam Tracking - PMC - PubMed Central, accessed February 1, 2026, <https://pmc.ncbi.nlm.nih.gov/articles/PMC12473589/>
  41. YOLOv8 model slowing down in real time detection after 2 hours - Ultralytics, accessed February 1, 2026, <https://community.ultralytics.com/t/yolov8-model-slowng-down-in-real-time-detection-after-2-hours/1373>
  42. Wrapping a C#/WPF GUI around C++/CLI around native C++ - Stack Overflow, accessed February 1, 2026, <https://stackoverflow.com/questions/6362434/wrapping-a-c-wpf-gui-around-c-c-li-around-native-c>
  43. Creating a C++/CLI Wrapper - Simple Talk - Redgate Software, accessed February 1, 2026, <https://www.red-gate.com/simple-talk/development/dotnet-development/creating-ccli-wrapper/>
  44. Tutorial - Ruckig, accessed February 1, 2026, <https://docs.ruckig.com/tutorial.html>
  45. kam3k/laser\_line\_extraction: A ROS package that extracts line segments from LaserScan messages. - GitHub, accessed February 1, 2026, [https://github.com/kam3k/laser\\_line\\_extraction](https://github.com/kam3k/laser_line_extraction)
  46. anderssandstrom/ruckig-ess: Instantaneous Motion Generation. Real-time. Jerk-constrained. Time-optimal. - GitHub, accessed February 1, 2026, <https://github.com/anderssandstrom/ruckig-ess>