🧩 Bối cảnh Ứng dụng Thực tế cho Pipeline V5

**🎯 1. Mục tiêu tổng quát của Pipeline**

Pipeline V5 được thiết kế để:

* **Tự động sinh timeline** cho từng robot trong dây chuyền mạ điện/mạ điện di.
* **Tối ưu hóa trình tự hoạt động** của các robot dựa trên input là:
  + **Route Step**: thứ tự các bước mà một rack đi qua.
  + **Timer**: thời gian xử lý tại từng vị trí (ví dụ: ngâm, xử lý, chờ...).

Nó mô phỏng **chu kỳ hoạt động lặp lại (CC)**, tương ứng với chu trình vận hành thực tế trên dây chuyền.

**📥 2. Input chính của hệ thống**

* **File Word (Route)**:  
  Mỗi dòng là một bước xử lý theo thứ tự từ nạp → xử lý → tháo, có thể chứa:
  + Marker vị trí (VD: 2D, 10T, 15D: timer)
  + Gán theo từng robot (R1, R2, R3)
  + Phân luồng (Luồng 1, Luồng 2...)
* **File Excel (Timer Config)**:  
  Chứa các cấu hình thời gian xử lý tiêu chuẩn (VD: 2D = 40s, 5D = 60s, v.v.), dùng cho bước áp timer (step 5).

**⚙️ 3. Pipeline xử lý ra gì?**

* Tạo **timeline chi tiết cho từng robot**, bao gồm:
  + Thời điểm nâng, hạ rack
  + Thời điểm delay (TDT)
  + Thời gian xử lý tại mỗi vị trí
  + Phát hiện chính xác điểm chuyển giao giữa các robot
* Xuất ra file:
  + .xlsx theo từng CC (timeline\_output\_cc1.xlsx, cc2, ...)
  + t\_den\_tracker.json theo dõi thời gian đến tại marker
  + Các file log kiểm soát chi tiết từng bước xử lý

**🧪 Tình huống sử dụng điển hình**

**🏭 Ứng dụng tại xưởng sản xuất có robot vận hành dây chuyền mạ:**

* Mỗi rack đi qua nhiều bể hóa chất và vị trí xử lý.
* Có các robot R1, R2, R3 phụ trách di chuyển rack giữa các bể.
* Kỹ thuật viên chỉ cần **nhập sơ đồ quy trình (file Word)** và **thời gian xử lý (Excel)** → pipeline sẽ:
  + **Tự động lập lịch cho robot**
  + **Cân bằng luồng xử lý**
  + **Bảo đảm không xung đột giữa các robot**

**🚀 Khả năng mở rộng ứng dụng trong tương lai**

**✅ 1. Chuyển đổi AutoCAD → Route tự động**

* Đọc layout CAD hoặc PDF → Trích xuất thành route\_step
* Không cần người kỹ thuật nhập file Word nữa → tự sinh dữ liệu đầu vào

**✅ 2. Kết hợp AI kiểm tra năng lực route**

* Một AI kiểm tra xem route có tối ưu hay không (xung đột, dư timer, delay dài…)
* Gợi ý cải tiến sơ đồ route trước khi chạy

**✅ 3. Tích hợp hệ thống thực tế (MES, SCADA)**

* Thay vì chỉ là mô phỏng, pipeline lấy trực tiếp thông tin từ hệ thống điều khiển thật
* Sinh timeline thực tế để điều phối robot theo thời gian thực (real-time)

Tổng Kết Logic Vận Hành Pipeline V5

# 1. Quy trình tổng thể từ khi chạy run\_pipeline\_local.py

* ➤ File: run\_pipeline\_local.py
* - Thiết lập đường dẫn, CC label, start\_time
* - Tự động nạp module
* - Gọi hàm: main\_pipeline\_v5.run\_full\_timeline\_pipeline(...)
* ➤ File: main\_pipeline\_v5.py
* - Đọc file .docx → sinh route\_steps
* - Gọi: run\_pipeline\_multi\_cc(...) trong v5\_loop\_controller
* ➤ File: v5\_loop\_controller\_v1\_2.py
* - Lặp từng CC theo 6 bước xử lý

# 2. Các bước trong mỗi CC

1. 1️⃣ Step 1 - Quét marker còn lại
2. 2️⃣ Step 2 - Tính timeline bán phần (R1, R2)
3. 3️⃣ Step 3 - Chọn marker tiếp theo cho R3
4. 4️⃣ Step 4 - Tính T\_đến tại marker
5. 5️⃣ Step 5 - Áp timer vào route
6. 6️⃣ Step 6 - Kiểm tra marker đã hoàn thành chưa

* → Nếu chưa xong thì lặp CC tiếp theo
* → Nếu xong thì kết thúc pipeline

# 3. Các module chức năng phụ trợ

* - timeline\_rb\_v4\_3\_1.py: Sinh timeline robot theo route
* - timeline\_exporter.py: Xuất timeline ra Excel
* - cc\_manager\_v1.py: Quản lý chu kỳ CC
* - write\_t\_den\_tracker.py: Ghi t\_den\_tracker.json
* - marker\_nap\_thao\_v2.py: Xử lý nạp/tháo hàng