

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA THÀNH PHỐ HỒ CHÍ MINH
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



**BÁO CÁO
ĐỒ ÁN TỐT NGHIỆP**

**PHÁT TRIỂN ỨNG DỤNG VR ĐIỀU KHIỂN CÁNH
TAY MÁY ROS JETARM**

NGÀNH: Kỹ thuật Máy tính

HỘI ĐỒNG: HỘI ĐỒNG 4 KỸ THUẬT MÁY TÍNH
GVHD: T.S LÊ TRỌNG NHÂN
GVPB: Th.S VŨ TRỌNG THIÊN

---o0o---

SVTH 1: NGUYỄN PHÚC TIẾN (2014725)
SVTH 2: NGUYỄN MINH TIẾN (2014722)
SVTH 3: NGUYỄN VĂN THỊNH (2014603)

TP. Hồ Chí Minh, 5/2024

LỜI CAM KẾT

Chúng tôi cam kết rằng Đồ Án này dựa trên ý tưởng và kiến thức của những người giám sát của chúng tôi. Tất cả các nghiên cứu và dữ liệu chưa được công bố. Các tài liệu tham khảo, các số liệu và thống kê là đáng tin cậy và trung thực. Nhóm đã hoàn thành các yêu cầu của Đồ án tốt nghiệp do khoa Khoa học và Kỹ thuật Máy tính đề ra.

Nguyễn Phúc Tiên

Nguyễn Minh Tiên

Nguyễn Văn Thịnh

LỜI CẢM ƠN

Trong thời gian thực hiện Đồ án tốt nghiệp, chúng em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô và bạn bè.

Đặc biệt, chúng em xin bày tỏ sự kính trọng và lòng biết ơn sâu sắc nhất đến thầy giáo hướng dẫn Đồ án của chúng em - TS. Lê Trọng Nhân, thầy là người đã tận tình hướng dẫn, chỉ bảo để chúng em có thể hoàn thành đồ án này. Thầy đã luôn đồng hành và sẵn sàng giúp đỡ chúng em mỗi khi cần, thầy cho chúng em động lực, nguồn cảm hứng và những gợi ý quý báu để có thể vượt qua những giai đoạn khó khăn trong quá trình thực hiện đồ án.

Chúng em cũng xin chân thành cảm ơn các thầy cô giáo trong Trường Đại học Bách khoa - Đại học Quốc gia TP.HCM nói chung, các thầy cô trong Khoa Khoa học và Kỹ thuật Máy tính nói riêng đã dạy dỗ cho chúng em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp chúng em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ chúng em trong suốt quá trình học tập.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án này không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để chúng em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Cuối cùng, chúng em xin kính chúc quý thầy, quý cô, quý nhà trường luôn luôn mạnh khỏe. Kính chúc cho chặng đường lái đò của quý thầy cô được thuận buồm xuôi gió!

Chúng em xin chân thành cảm ơn.

TỔNG QUAN

Trong những năm gần đây, sự phát triển của giao tiếp Machine-to-Machine (M2M) và Internet of Things (IoT) trong Công nghiệp đã mở đường cho việc phát triển các máy móc thông minh, đặc biệt là trong lĩnh vực sản xuất, với sự gia tăng đáng kể của quá trình tự động hóa và robot hóa. Tuy nhiên, việc thay thế lao động bằng robot đã làm giảm khả năng giám sát của người vận hành và quản lý, và để giải quyết vấn đề này, cần sự hỗ trợ từ hình ảnh thực tế trực tiếp từ nơi làm việc.

Trong ngữ cảnh này, nhóm nghiên cứu của chúng tôi đề xuất phát triển một ứng dụng thực tế ảo (VR), đặc biệt kết hợp với hệ thống 3 camera, để điều khiển cánh tay robot. Việc tích hợp VR với 3 camera giống như một môi trường 3D ngoài đời thực mang lại những trải nghiệm giám sát độc đáo và chi tiết. Điều này không chỉ giúp người quản lý và vận hành cánh tay robot có cái nhìn toàn diện về môi trường làm việc mà còn tạo ra sự hiểu biết sâu sắc về các quy trình tự động hóa.

Đồng thời, chúng tôi cũng giải quyết vấn đề về truyền tải hình ảnh trực tiếp từ nơi làm việc bằng cách sử dụng giao thức RTSP (Real Time Streaming Protocol). Việc này không chỉ đảm bảo việc truyền tải dữ liệu một cách hiệu quả và nhanh chóng mà còn tạo ra một liên kết trực tiếp giữa thế giới thực và môi trường ảo trong ứng dụng VR.

Chúng tôi muốn mô phỏng chuyển động của cánh tay robot ở ngoài thực tế lên ứng dụng thực tế ảo thông qua kỹ thuật digital twin. Kết hợp với các camera, chúng ta có thể quan sát trực tiếp được chuyển động, sự vận hành của cánh tay trong ứng dụng và khung cảnh xung quanh cánh tay robot để có cái nhìn trực quan nhất cho người sử dụng.

Vì vậy, Đồ án này sẽ tập trung vào nghiên cứu và hiện thực **Phát triển ứng dụng VR điều khiển cánh tay máy ROS JETARM**. Chúng tôi tin rằng giải pháp này sẽ không chỉ giúp cải thiện hiệu suất quản lý mà còn đưa ra một trải nghiệm giám sát thực tế và hiệu quả trong môi trường sản xuất tự động hóa.

Các từ khóa : Digital Twin, Robot Arm 6DoF, VR/ AR application, RTSP, MQTT, M2M, IoT, Công nghiệp 4.0.

MỤC LỤC

Lời cam kết	ii
Lời cảm ơn	iii
Tổng quan	iv
Danh sách hình vẽ	x
Danh sách bảng	xv
1 Giới thiệu	1
1.1 Tổng quan về Công nghiệp 4.0	2
1.2 Nhà máy thông minh - Smart factory	5
1.2.1 Định nghĩa và lợi ích của Smart Factory	5
1.2.2 Kiến trúc của Smart Factory	6
1.3 Sản xuất thông minh trong thời đại công nghệ 4.0.	7
1.4 Cánh tay Robot trong sản xuất thông minh.	11
1.4.1 Khái niệm	11
1.4.2 Cấu tạo của cánh tay Robot.	12
1.4.3 Các ứng dụng của VR trong điều khiển cánh tay Robot	
13	

1.5	Công nghệ VR/AR	16
1.5.1	Khái niệm công nghệ VR	16
1.5.2	Khái niệm công nghệ AR	17
1.5.3	Các ứng dụng trong công nghệ VR/AR	18
1.5.4	Ứng dụng trong lĩnh vực robot	18
1.6	Giới thiệu về đồ án	23
1.7	Bô cục của đồ án	25
2	Cơ sở lý thuyết	26
2.1	Digital Twin	27
2.1.1	Khái niệm về Digital Twin	27
2.1.2	Những nhầm lẫn về Digital Twin	28
2.1.3	Ứng dụng của Digital Twin	29
2.2	Tổng quan về các giao thức M2M	32
2.2.1	Giới thiệu về M2M	32
2.2.2	Nguyên lý hoạt động của M2M	32
2.2.3	Ứng dụng của M2M	33
2.2.4	MQTT	34
2.2.5	RTSP	40
2.2.6	HTTP	43
2.3	Toạ độ và góc trong không gian	48
2.3.1	Toạ độ Euler	48

2.3.2	Toạ độ Quaternion	52
2.3.3	Toạ độ hướng và góc nghiêng của Camera	54
3	Thiết kế và hiện thực hệ thống	56
3.1	Tổng quan hệ thống.	57
3.1.1	Kiến trúc hệ thống	57
3.1.2	Giới thiệu thiết bị và công nghệ	58
3.2	Ứng dụng thực tế ảo (VR Application)	67
3.2.1	Cách tổ chức chương trình VR application	68
3.2.2	Hiện thực module VR application	69
3.2.3	Hiện thực chức năng liên quan đến Camera	72
3.2.4	Hiện thực chức năng liên quan lên quan giao tiếp MQTT giữa ứng dụng VR và cánh tay robot	76
3.2.5	Hiện thực về giao diện ứng dụng.	78
4	Đánh giá kết quả	83
4.1	Kết quả đạt được	83
4.1.1	VR Application	84
4.2	Kiểm tra đánh giá hệ thống	92
4.2.1	Về thời gian cập nhật mỗi gói tin trong trên mỗi frame ảnh trong việc Streaming	92
4.2.2	Về sự thay đổi về thông số trong Unity	94
4.2.3	Số khung hình trên giây của ứng dụng Oculus.	98
4.2.4	Về thời gian trễ của giao thức MQTT	99

4.2.5 Về thời gian trễ khung hình khi stream rtsp	100
5 TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN	101
5.1 Tổng kết.	101
5.2 Hướng phát triển tương lai	103
Tài liệu tham khảo	105

DANH SÁCH HÌNH VẼ

1.1	Kiến trúc chung cho các ứng dụng thời đại 4.0 [5]	3
1.2	Các công nghệ của Công nghiệp 4.0	8
1.3	Ảnh mô tả về cánh tay Robot	12
1.4	Cánh tay robot xếp chồng hàng hóa [7]	14
1.5	Cánh tay robot xác định vật thể, đối tượng [3]	14
1.6	Cánh tay robot sắp xếp các vật thể theo màu [1]	15
1.7	Cánh tay robot gấp và thả hàng hóa [7]	16
1.8	Ảnh mô tả về thực tế ảo	16
1.9	Ảnh mô tả về thực tế ảo tăng cường [32]	18
1.10	Sử dụng VR trong lĩnh vực robot [8]	19
1.11	Sử dụng VR trong chiến dịch quảng cáo và marketing [39]	19
1.12	Sử dụng VR trong việc xem phim giải trí [39]	20
1.13	Sử dụng VR trong ngành du lịch [39]	21
1.14	Sử dụng VR trong quá trình đào tạo bác sĩ [39]	22
1.15	Sử dụng VR trong quá trình thiết kế xe hơi [39]	22
1.16	Sử dụng VR trong giáo dục [39]	23
2.1	Một số khái niệm nhầm lẫn của Digital Twin	28

2.2	Kiến thức chung của MQTT sử dụng MQTT Broker	35
2.3	Ba mức Chất lượng Dịch vụ (QoS)	36
2.4	Cấu trúc của topic [40]	37
2.5	Cách thức hoạt động của RTSP [45]	41
2.6	HTTP request [6]	45
2.7	HTTP response [6]	46
2.8	Hình ảnh về trực roll, pitch, yaw trong không gian 3D [14]	48
2.9	Vị trí ban đầu của vật thể theo góc Euler [14]	49
2.10	Xoay vật thể theo trục y [14]	49
2.11	Xoay vật thể theo trục x trong không gian đối tượng [14] . .	50
2.12	Xoay vật thể theo trục z trong không gian đối tượng [14] . .	50
2.13	Hình ảnh minh họa về gimbal lock	52
2.14	Ví dụ về phép quay quaternion trong không gian [28] . . .	54
2.15	Ví dụ về góc azimuth và elevation [46]	55
3.1	Kiến trúc ba lớp gồm virtual (lớp ảo), network (mạng), và physical (vật lý). Lớp virtual tương tác với người dùng và cung cấp trải nghiệm thực tế ảo. Lớp network quản lý giao tiếp giữa các thành phần ảo và vật lý trong hệ thống. Cuối cùng, lớp physical bao gồm các thiết bị và tài nguyên vật lý như mô hình robot, jetson nano và ipcamera.	57
3.2	Hình ảnh Jetson Nano	59
3.3	Hình ảnh cánh tay Robot Arm 6DoF ngoài đời thật	60
3.4	Hình ảnh cánh tay Robot Arm 6DoF ngoài đời thật	61
3.5	Hình ảnh Oculus Quest 2	62

3.6	Hình ảnh 3 camera	63
3.7	Hình ảnh ba camera được lắp đặt thực tế	64
3.8	OpenCV	66
3.9	Unity3D	67
3.10	Cấu trúc thư mục của VR Application trong Unity 3D	69
3.11	Sơ đồ khối của VR app	70
3.12	Chú thích chức năng điều khiển sử dụng tay cầm Oculus Quest 2	71
3.13	Sơ đồ kết nối mạng cục bộ với camera	73
3.14	Tạo GameObject và cấu hình để hiển thị nội dung trực tiếp trong Unity 3D	73
3.15	Sequence diagram gửi request xoay đầu của kính VR của camera Hikvision	74
3.16	Sequence diagram stream RTSP từ webcam robot tới môi trường Unity	75
3.17	Trang thái tương tác ứng dụng VR và cánh tay Robot	76
3.18	Sequence diagram của chức năng điều khiển cánh tay theo servo	77
3.19	Sequence diagram stream RTSP từ webcam robot tới môi trường Unity	78
3.20	Mô hình 3D của cánh tay trong Blender	79
3.21	Mô hình 3D hoàn chỉnh của cánh tay trong Blender	79
3.22	Mô hình 3D của cánh tay trong Unity 3D	80
3.23	Mô hình 3D của cánh tay trong Unity 3D	81
4.1	Giao diện ứng dụng khi vừa khởi động	84

4.2 Giao diện ứng dụng bên góc trái của người sử dụng hiển thị debug log trong chương trình unity	85
4.3 Giao diện ứng dụng bên góc phải của người sử dụng hiển thị camera của robot và cánh tay robot ảo	85
4.4 Giao diện ứng dụng khi ở mục Connection	86
4.5 Giao diện khi kết nối thành công với cánh tay robot	87
4.6 Giao diện khi kết nối thất bại với cánh tay robot	87
4.7 Giao diện bắt đầu của bảng điều khiển với manaul mode .	88
4.8 Giao diện của bảng điều khiển với Auto mode	88
4.9 Giao diện Menu con của chức năng Chinese Chess	89
4.10 Nhận dạng bàn tay và hiển thị lên kính VR	90
4.11 Hình ảnh thông số ở trạng thái mặc định, không có khớp nào được quay	91
4.12 Hình ảnh thông số của khớp thứ 2 ở hai trường hợp gần tới ngưỡng và đạt ngưỡng giới hạn	91
4.13 Hình ảnh thông số của khớp thứ 3 ở hai trường hợp gần tới ngưỡng và đạt ngưỡng giới hạn	92
4.14 Các gói tin cập nhật frame	93
4.15 Các gói tin cập nhật frame - tiếp theo	93
4.16 Hình ảnh biểu diễn trạng thái Static khi chưa Run trong Unity	95
4.17 Hình ảnh biểu diễn trạng thái Static khi đã Run trong Unity	95
4.18 Phân tích các thành phần trong Unity	96
4.19 Phân tích các thành phần trong Unity	96
4.20 Phân tích các thành phần trong Unity	97
4.21 Log thể hiện thời gian update mỗi frame	98

4.22 Số khung hình trên giây (FPS) khi không đeo kính và khi đeo kính Oculus	99
4.23 Độ trễ của giao thức MQTT	100
4.24 Độ trễ khi lấy luồng video bằng RTSP trên unity	100

DANH SÁCH BẢNG

1.1 Ứng dụng của các công nghệ Công nghiệp 4.0	9
--	---

1

GIỚI THIỆU

Chương này cung cấp một cái nhìn tổng quan về các ứng dụng thực tế ảo khi áp dụng vào cánh tay Robot. Sự bùng nổ của cách mạng Công nghệ 4.0 đã tạo tiền đề cho sự phát triển nhanh chóng của công nghệ điện - điện tử, công nghệ thông tin và công nghệ sản xuất tiên tiến, phương thức sản xuất của các doanh nghiệp sản xuất đang được chuyển từ kỹ thuật số sang thông minh. Và Nhà máy thông minh (Smart Factory) là một giải pháp sản xuất linh hoạt và hiệu quả để đáp ứng được nhu cầu của thị trường ngày nay. Cánh tay Robot đã trở thành một phần không thể thiếu trong các nhà máy thông minh, nhằm mục đích vận hành chính xác, nâng cao năng suất và thay thế sức người trong nhiều công đoạn nặng nề của quy trình sản xuất. Vai trò của con người lúc này, trở thành người quản trị, giám sát và điều khiển từ xa các cánh tay Robot thông qua các công nghệ kết nối Internet of Things (IoTs) hiện đại mà cách mạng công nghệ mang lại. Không chỉ các thông số của cánh tay Robot, mà hình ảnh trực tiếp từ nó cũng có thể được gửi về người giám sát một cách thuận lợi. Những tiền đề này, mở ra các cơ hội cho các ứng dụng dựa trên thực tế ảo. Những thuận lợi và thách thức khi đưa ứng dụng dựa trên kính VR để điều khiển cánh tay Robot cũng sẽ được phân tích trong chương này.

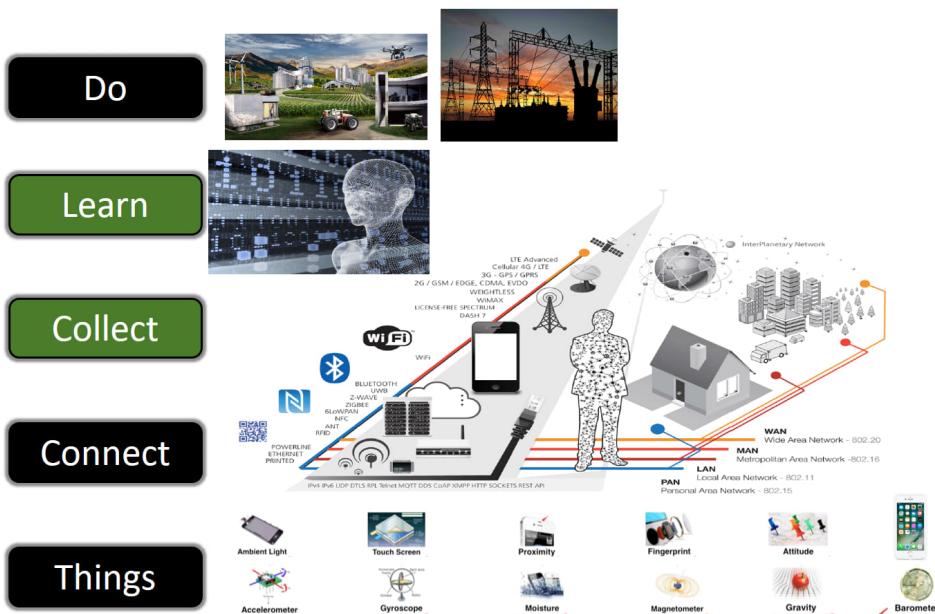
1

1.1. TỔNG QUAN VỀ CÔNG NGHIỆP 4.0

Internet vạn vật, hay còn gọi là Internet of Things – IoTs, là một cuộc cách mạng trong việc kết nối giữa các thiết bị không dây với nhau. Ban đầu, chúng ta có mạng Internet, một thành tựu của cuộc cách mạng khoa học công nghệ lần thứ 3, cho phép các máy tính có thể kết nối và trao đổi thông tin toàn cầu. Tuy nhiên, với sự phát triển nhanh chóng của ngành vi cơ điện tử (Micro Electro Mechanical System) [36], không chỉ máy tính, giờ đây rất nhiều các thiết bị có khả năng kết nối vào mạng Internet. Thông dụng nhất trong cuộc sống mà chúng ta có thể kể đến như các điện thoại thông minh, máy tính bảng, các loại thẻ thông minh (Smart cards) hay như các nốt trong mạng cảm biến không dây (Wireless Sensor Networks). Với những đặc tính đó, một thế hệ mạng mới đã được hình thành, và là sản phẩm đặc trưng cho cuộc cách mạng khoa học công nghệ lần thứ 4 [43], mạng Internet vạn vật, hay còn gọi là IoT - Internet of Things.

Dựa trên mạng Internet vạn vật, các ứng dụng không còn ở khái niệm thông minh nữa, mà sẽ tiến lên một bước cao hơn, gọi là tự hành (autonomous), chẳng hạn như các ứng dụng giám sát và tự động thích nghi trong việc điều khiển như các dịch vụ trong nhà, bãi giữ xe [2], hay các hệ thống quan trắc trong nông nghiệp, thủy hải sản. Theo diễn giả nổi tiếng Timothy Chou, kiến trúc của các ứng dụng thời đại cách mạng công nghiệp 4.0 dựa trên Internet vạn vật nói chung và sản xuất thông minh trong công nghiệp nói riêng, được chia thành mô hình 5 lớp, như mô tả ở hình bên dưới.

Lớp đầu tiên bao gồm các Things. Things ở đây chính các máy móc và chúng được kết nối với Internet theo nhiều cách khác nhau. Sau khi được kết nối, Lớp Collect ở đây đề cập đến các công nghệ được thiết kế để thu thập dữ liệu - dữ liệu chuỗi thời gian được gửi đi mỗi giờ, phút hoặc giây. Lớp thứ tư là Learn. Không giống như trong thế giới của các ứng dụng IoP (Internet of People), nơi chúng ta phải nhập những nội dung input nào đó, thì các ứng dụng IoT sẽ lấy dữ liệu một cách liên tục. Ví dụ, chúng ta có thể sử dụng máy móc để học hỏi (learn) từ những Things của mình tại bệnh viện, hầm mỏ hoặc trang trại. Và cuối cùng, chúng ta sẽ có những câu hỏi, Những công nghệ này để làm gì? Kết quả kinh doanh là gì? Thì khi đó, Lớp Do mô tả cả công nghệ ứng dụng phần mềm và mô hình kinh doanh bị ảnh hưởng bởi các công ty sản xuất ra Things, cũng như những người sử dụng chúng để cung cấp dịch vụ chăm sóc sức khỏe, vận



Hình 1.1: Kiến trúc chung cho các ứng dụng thời đại 4.0 [5]

chuyển hoặc xây dựng chẵng hạn. Và chức năng chính của từng lớp trong kiến trúc này được khái quát như sau:

- **Things:** Các thiết bị trong ứng dụng giám sát. Chúng ta có thể thấy, đây là lớp rất phong phú về mặt số lượng và đa dạng về chức năng. Rất nhiều các loại cảm biến sẽ được dùng, tùy vào các ứng dụng giám sát. Bên cạnh đó, các nốt cảm biến sẽ chủ yếu dựa vào giao tiếp không dây
- **Connect:** Thu thập dữ liệu từ các nốt cảm biến. Do có rất nhiều tiêu chuẩn kết nối tùy theo từng loại ứng dụng, lớp này phải hỗ trợ nhiều loại kết nối, từ giao tiếp Zigbee và Wifi trong các ứng dụng nhà thông minh, với khoảng cách giao tiếp ngắn cho đến các giao trên không gian rộng như LoRa hay 3G/4G.
- **Collect:** Sau khi dữ liệu được thu thập, chúng sẽ được gửi lên các server tập trung để lưu trữ dữ liệu. Tại đây, một lượng lớn dữ liệu sẽ được đẩy về, tạo ra một thách thức không nhỏ cho các server và phải ứng dụng các công nghệ về Big Data (dữ liệu lớn) để xử lý.
- **Learn:** Nhiệm vụ của lớp này là lọc ra các thông tin đặc trưng, có

1

ngữ nghĩa đặc thù cho từng loại ứng dụng. Các công nghệ về Học Máy và hiện tại là Học Sâu (Deep Learning) sẽ được áp dụng ở đây.

- **Do:** Dựa vào các thông tin đặc trưng, hệ thống sẽ xây dựng nên những quy luật thích nghi theo ngoại cảnh, và đề xuất các quyết định cho hệ thống. Với mỗi quyết định, việc thực thi sẽ được đo đạc một cách tự động, và sai lệnh của quyết định đó so với mục tiêu tối ưu sẽ được xem xét lại cho lần sau. Theo cách này, hệ thống sẽ tự động tích lũy “kinh nghiệm” trong một thời gian dài, để ngày càng trở nên thông minh và hoàn thiện hơn.

Sự phát triển mạnh mẽ của khoa học công nghệ ở những năm đầu thế kỉ 21 đã đưa nhân loại bước vào cách mạng công nghiệp 4.0, khi mà "hàng tỉ" thiết bị có thể giao tiếp và chia sẻ dữ liệu với nhau thông qua mạng kết nối vạn vật (Internet of Things). Theo ước tính của cộng đồng khoa học, đến năm 2025 sẽ có 75 tỉ thiết bị có thể kết nối mạng Internet với nhau [4]. Và với sự bùng nổ về số lượng thiết bị kết nối đó, yêu cầu về sự phát triển của các lớp trong mô hình là cần thiết, việc đáp ứng sự tương tác giữa các thiết bị, cảm biến, việc xây dựng hệ thống hạ tầng và bảo mật, ... là cần thiết, việc mô phỏng và đảm bảo an toàn hạn chế tối đa rủi ro khi ứng dụng vào chuỗi sản xuất công nghiệp là một vấn đề cần được giải quyết.

1.2. NHÀ MÁY THÔNG MINH - SMART FACTORY

1

Dưới sự ảnh hưởng của cách mạng công nghiệp 4.0 và sự ra đời, phát triển của vô số các công nghệ mới như IoT, IIoT, Cloud computing, Cyber-security, Big Data Analytics... như trình bày ở trên, đã dẫn đến sự thay đổi lớn cho xu hướng công nghiệp thời đại mới, chú trọng hơn vào dữ liệu, về sự kết nối và tính thông minh, các mô hình sản xuất thuộc nhiều lĩnh vực khác nhau đã có những bước thay đổi, phát triển để phù hợp hơn, mà trong đó không thể không kể đến là mô hình nhà máy thông minh (Smart Factory). Smart Factory là một mô hình sản xuất công nghiệp đánh dấu bước trưởng thành đột phá của sản xuất trong thời đại công nghiệp 4.0 so với thời kỳ 3.0 trước đó, đem lại định nghĩa mới về sản xuất tại nhà máy không chỉ đơn thuần là "tự động", mà còn phải "tự hành" và "thông minh". Để có thể xây dựng một mô hình Smart Factory, ta có thể áp dụng và kết hợp các công nghệ mới của thời đại 4.0 trên, và đảm bảo chúng cần được liên kết với nhau để tận dụng tối đa sức mạnh mà các công nghệ mang lại. Trong phạm vi đề tài, nhóm chỉ giới hạn tìm hiểu về ứng dụng của công nghệ IoT, cụ thể là IIoT (Industrial Internet of Things) trong bối cảnh nhà máy thông minh.

Phần tiếp theo, nhóm sẽ trình bày hai ý chính của smart factory: Khái niệm và lợi ích của smart factory, kiến trúc đề xuất của nhóm cho các ứng dụng Smart Factory dựa trên mô hình năm lớp của diễn giả Timothy Chou.

1.2.1. ĐỊNH NGHĨA VÀ LỢI ÍCH CỦA SMART FACTORY

Smart Factory (Nhà máy thông minh) là một cơ sở sản xuất được "số hóa" sử dụng các thiết bị, máy móc và hệ thống sản xuất kết nối với nhau nhằm liên tục thu thập thông tin và chia sẻ dữ liệu. Dữ liệu này được sử dụng để tạo ra những dự đoán nhằm cải thiện quá trình sản xuất và chỉ ra các vấn đề đang gặp phải trong tiến trình này. Định nghĩa trên, ta có thể chỉ ra hai ý chính làm nên một Smart Factory, đó chính là "sự kết nối của tất cả thiết bị" và "dự đoán". Thực vậy, nói đến smart factory, không phải ý chỉ là nhà máy chỉ trang bị những máy móc hiện đại và tự động là đủ, bởi những cỗ máy "tự động" này đã xuất hiện từ lâu ở thời đại công nghiệp 3.0. Sự khác biệt giữa Smart factory trong thời đại công nghiệp 4.0 này chính là việc các máy móc thay vì độc lập với nhau, chúng sẽ thông minh hơn, biết kết nối, giao tiếp và chia sẻ thông tin, tạo thành

1

một mạng lưới thông tin khổng lồ giữa các thiết bị trong nhà máy. Hơn nữa, với lượng dữ liệu khổng lồ như vậy, sẽ là cơ hội để ta thu thập và phân tích chúng - sử dụng trí tuệ nhân tạo để có thể thấy rõ những đặc trưng của sản xuất một cách thực tế, đồng thời giúp ta dự đoán những sự kiện mới, thậm chí là tạo ra những cải tiến cho mô hình sản xuất và truyền lại để các máy móc áp dụng. Smart Factory đã đề ra một nhà máy "kết nối" và "thông minh" hơn so với sự tự động hóa đơn thuần của các nhà máy thời đại công nghiệp 3.0

1.2.2. KIẾN TRÚC CỦA SMART FACTORY

Trong thời đại công nghiệp 4.0, sản xuất thông minh (intelligent manufacturing) thu hút nhiều sự quan tâm từ nhà nước, các chuyên gia và các nhà nghiên cứu khoa học. Theo đó, sự xây dựng nền kiến trúc của Smart Factory cũng được nghiên cứu. Smart Factory là dựa trên phương hướng xây dựng nhà máy "số hóa" (digital) và "tự động" (autlomated), sử dụng công nghệ thông tin như nền tảng đám mây hoặc IIoT để cải thiện khả năng quản lý tài nguyên sản xuất và chất lượng dịch vụ. [9] Hiện tại, vẫn chưa có một hiện thực nào của Smart factory được chuẩn hóa và công bố, tuy nhiên, dựa theo nhiều nghiên cứu khác nhau, Chen et al. [9] cho rằng có ba lớp chính trong kiến trúc của Smart factory:

- **Lớp vật lý:** gồm các thiết bị vật lý cần sự hỗ trợ để lấy được dữ liệu nhanh chóng trong thời gian thực, các thiết bị giao tiếp phải cung cấp khả năng truyền nhanh chóng, kể cả loại dữ liệu phức tạp.
- **Lớp mạng:** Lớp này quy định cách thức kết nối giữa các thiết bị vật lý và thu thập chúng tới các server edge để xử lý. Về phương thức kết nối, smart factory sử dụng Industrial Internet of Thing - IIoT. Với việc áp dụng IIoT, Các phương pháp để giao tiếp giữa các thiết bị trong nhà máy như Mạng cảm biến công nghiệp (Industrial Wireless Sensor Networks - IWSNs) cần được nghiên cứu.
- **Lớp dữ liệu** Dữ liệu thu thập được từ các thiết bị cần được tập hợp ở nền tảng đám mây, sau đó có khả năng phân tích tìm ra ngữ nghĩa của nhiều loại dữ liệu khác nhau, đặc biệt hơn hết là tạo ra khả năng tự tổ chức, tự học và tự áp dụng ở các thiết bị thông minh này nhằm cải thiện năng suất nhà máy. Đây là lớp áp dụng các công nghệ phân tích, trí tuệ nhân tạo nhằm tận dụng lượng dữ liệu khổng lồ thu thập

được từ các thiết bị để đưa ra những dự đoán và tối ưu cho công tác sản xuất của smart factory

Có thể thấy, ba lớp kiến trúc được trình bày trên có những điểm tương đồng nhất định với kiến trúc năm lớp cho các ứng dụng thời đại 4.0 được đề xuất bởi diễn giả Timothy Chou: Lớp vật lý gồm "Things", lớp mạng gồm "Connect" và "Collect", và lớp dữ liệu bao gồm "Learn" và "Do". Mô hình năm lớp của diễn giả Chou có sự phân tách sâu hơn về chức năng của một ứng dụng 4.0, do đó sẽ được nhóm tập trung phân tích. Trong phạm vi của đề tài, nhóm sẽ chú trọng vào lớp vật lý (Things) trong ứng dụng thời đại 4.0 nói chung và trong kiến trúc ứng dụng smart factory nói riêng, nhằm có cái nhìn cụ thể hơn về việc giao tiếp giữa các thiết bị vật lý về việc thu thập thông tin không đồng nhất và xử lý chúng một cách nhanh chóng.

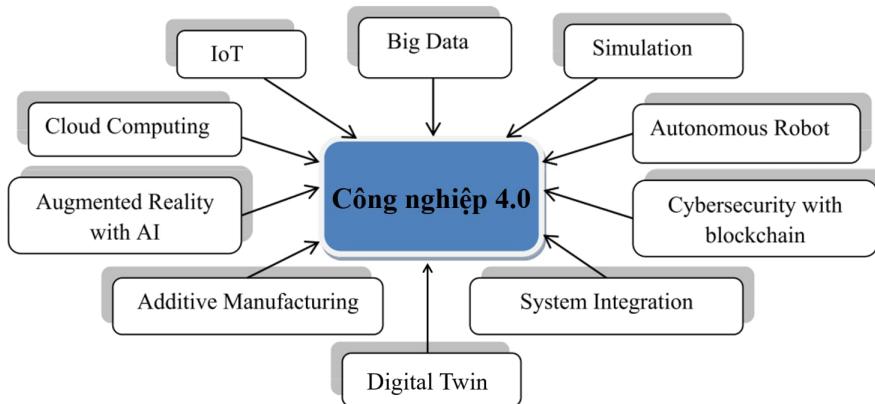
Ở phần tiếp theo, nhóm sẽ tập trung vào lớp vật lý này, nhằm nhìn nhận những thách thức trong việc hiện thực lớp vật lý cho ứng dụng smart factory và đề ra những giải pháp đang được áp dụng hiện nay để giải quyết những thách thức đó

1.3. SẢN XUẤT THÔNG MINH TRONG THỜI ĐẠI CÔNG NGHỆ 4.0

Ứng dụng công nghệ Công nghiệp 4.0 bao gồm các công nghệ khác nhau như Internet vạn vật (IoT), điện toán đám mây, additive manufacturing, an ninh mạng với blockchain, augmented reality với trí tuệ nhân tạo (AI), big data, tích hợp hệ thống (system integration), mô phỏng (simulation), robot tự hành. Nhiều công nghệ trong số này đã được sử dụng trong sản xuất, nhưng với ngành công nghiệp 4.0, chúng sẽ chuyển đổi thành quy trình sản xuất dưới dạng các cell được tối ưu hóa sẽ kết hợp với nhau thành một quy trình sản xuất được tích hợp hoàn toàn, tự động hóa và tối ưu hóa. Chúng làm tăng hiệu quả và thay đổi hệ thống chuỗi cung ứng truyền thống, cũng như mối quan hệ giữa con người và máy móc. Các kỹ thuật của Công nghiệp 4.0 có khả năng cải thiện việc sử dụng năng lượng, thiết bị và nguồn nhân lực. Công nghiệp 4.0 là một cấu trúc tương lai nuôi dưỡng sự phát triển của các hệ thống sản xuất tự trị với ứng dụng IoT, CPS và AI. Các công nghệ dựa trên cảm biến mới giúp các doanh nghiệp vừa và nhỏ theo dõi liên tục việc sử dụng máy móc, nhu cầu năng

1

lượng và đào tạo nhân viên. Bằng cách phân tích kỹ lưỡng các công nghệ Công nghiệp 4.0 khác nhau, dữ liệu từ các thiết bị IoT khác nhau có thể được xử lý để cải thiện tính bền vững của hoạt động sản xuất.



Hình 1.2: Các công nghệ của Công nghiệp 4.0

Các công nghệ này có thể đóng góp đáng kể cho sự bền vững bằng cách giảm lượng khí thải carbon, sử dụng năng lượng tái tạo và các giải pháp công nghệ phù hợp cho cả cá nhân và xã hội. Sự phát triển của Công nghiệp 4.0 giúp sử dụng tối ưu các nguồn tài nguyên theo cách minh bạch hơn. Bằng cách thực hiện các phương pháp của Công nghiệp 4.0, hiệu quả sản xuất và đổi mới có thể được cải thiện, ảnh hưởng đến tính bền vững về xã hội và môi trường. Có nhiều lợi ích được đóng góp riêng bởi các công nghệ khác nhau của Công nghiệp 4.0. Các ứng dụng quan trọng của các công nghệ Công nghiệp 4.0 này được tóm tắt trong Bảng 1.1

Bảng 1.1: Ứng dụng của các công nghệ Công nghiệp 4.0

Các công nghệ của Công nghiệp 4.0	Ứng dụng
IoTs	IoT tích hợp các cảm biến vào hệ thống sản xuất. Các cảm biến và máy móc của nhà sản xuất được kết nối mạng và sử dụng điện toán nhúng. Nó tạo ra khả năng thu thập và phân tích dữ liệu theo cách phi tập trung.
Điện toán đám mây (Cloud computing)	Truy cập dữ liệu có sẵn mọi lúc mọi nơi; tính minh bạch và khả năng đáp ứng của chuỗi cung ứng; dễ dàng chia sẻ dữ liệu quan trọng; hỗ trợ từ các đối tác trong chuỗi cung ứng trong việc nâng cấp công nghệ, dữ liệu về vòng đời sản phẩm có thể được lưu trữ và truy xuất theo triết lý CE.
Augmented Reality	Các hệ thống dựa trên Augmented Reality hỗ trợ nhiều dịch vụ khác nhau, ví dụ như khả năng hiển thị thông số của robot trực tiếp trong thế giới ảo giúp tối ưu hóa giám sát, giảm thời gian phản ứng và nâng cao khả năng quản lý từ xa, tạo ra môi trường làm việc linh hoạt và hiệu quả. AR không chỉ là công nghệ hỗ trợ, mà là công cụ quan trọng giúp tối ưu hóa hiệu suất và đào tạo robot theo nhu cầu đặc biệt của sản xuất trong môi trường công nghiệp 4.0.

1

Bảng 1.1 – Tiếp tục từ trang trước

Các công nghệ của Công nghiệp 4.0	Ứng dụng
Sản xuất phụ gia (Additive manufacturing - AM)	Sản xuất phụ gia hoặc in 3D được sử dụng để sản xuất các lô sản xuất nhỏ theo yêu cầu. Nó tạo ra những lợi thế như là: sản xuất các sản phẩm phức tạp và cấu trúc nhẹ. Hệ thống AM phi tập trung và hiệu suất cao này sẽ giảm khoảng cách vận chuyển và lượng hàng tồn kho. Do đó, nó sửa đổi hệ thống chuỗi cung ứng.
Tích hợp hệ thống (System integration)	Trong ngành công nghiệp 4.0, các công ty, bộ phận, phòng ban, chức năng sẽ trở nên gắn kết hơn nhiều, với tư cách là cross-company. Do đó, cross-company này, cũng như các mạng tích hợp dữ liệu toàn cầu phát triển và cho phép các chuỗi giá trị thực sự tự động. Do đó, nó tạo ra sự kết nối trong chuỗi cung ứng, giữa nhà cung cấp và khách hàng thông qua tích hợp theo chiều ngang và dọc.
An ninh mạng (Cyber-security)	Công nghiệp 4.0 làm tăng khả năng kết nối và sử dụng các giao thức truyền thông tiêu chuẩn. Khi đó, cần phải bảo vệ các hệ thống thông tin, hệ thống công nghiệp, dây chuyền sản xuất và thiết bị khỏi các mối đe dọa an ninh mạng với tần suất ngày càng tăng cao. Điều cần thiết là tạo ra thông tin liên lạc an toàn, đáng tin cậy, cũng như quản lý truy cập và nhận dạng tinh vi của máy móc và người dùng.

Bảng 1.1 – Tiếp tục từ trang trước

Các công nghệ của Công nghiệp 4.0	Ứng dụng
Robot tự hành	Ngày nay, robot được sử dụng để giải quyết các nhiệm vụ phức tạp và cộng tác với nhau và với con người. Những robot này tự chủ, linh hoạt và hợp tác. Hơn nữa, chúng có chi phí thấp hơn và có nhiều khả năng hơn so trước đây.
Mô phỏng (Simulation)	Mô phỏng 2D hoặc 3D của quá trình phát triển sản phẩm, phát triển vật liệu và sản xuất đã được sử dụng trong thế giới công nghiệp. Ngày nay, mô phỏng phải được sử dụng rộng rãi hơn trong các hoạt động của nhà máy. Những mô phỏng này cung cấp dữ liệu thời gian thực để phản ánh thế giới thực trong một mô hình ảo, bao gồm máy móc, sản phẩm và con người. Nó giúp làm giảm thời gian thiết lập máy và tăng chất lượng sản xuất.
Phân tích dữ liệu lớn (Big Data Analytics)	Dữ liệu được thu thập từ nhiều thiết bị dựa trên IoT có thể được phân tích để lấy thông tin & xu hướng; dữ liệu có thể được sử dụng để lập trình các thiết bị AI; việc sử dụng máy móc và nguồn nhân lực có thể được tối ưu hóa; khả năng truy xuất nguồn gốc của sản phẩm sẽ cải thiện.

1.4. CÁNH TAY ROBOT TRONG SẢN XUẤT THÔNG MINH

1.4.1. KHÁI NIỆM

Cánh tay Robot công nghiệp (Cánh tay cơ khí) là một thiết bị được lập trình để hoạt động tương tự như cánh tay con người, với các khớp chuyển

1

động theo một trục dọc và có thể xoay theo các hướng nhất định.

Hầu hết các robot công nghiệp được chế tạo và lập trình để thực hiện các nhiệm vụ cụ thể. Kích cỡ của chúng có thể nhỏ để cầm nắm để thực hiện các thao tác phức tạp, hoặc đủ lớn để nâng nhắc khôi lượng nặng.

Cánh tay robot lần đầu tiên được ứng dụng trong ngành công nghiệp ô tô với công việc hàn lắp đi lắp lại. Đến nay, nhờ sự phát triển của công nghệ nó đã góp mặt trong nhiều lĩnh vực: cơ khí, y tế, thực phẩm, hàng không vũ trụ, ...

1.4.2. CẤU TẠO CỦA CÁNH TAY ROBOT

Ở đề tài này, nhóm chúng tôi sử dụng cánh tay Robot được cấu tạo bởi 3 phần như sau:



Hình 1.3: Ảnh mô tả về cánh tay Robot

1. Tay máy (Phần cứng)

Phần tay máy thường được chế tạo từ gang và thép – những vật liệu có độ bền cao. Tay máy được mô phỏng như cánh tay người, có phần cổ tay, cẳng tay, khuỷu tay, vai và chân đế. Các cánh tay robot

trong công nghiệp có từ 4 – 6 khớp nối với 6 bậc tự do tương đương với 6 cách di chuyển khác nhau.

2. Hệ thống điều khiển

Chịu trách nhiệm điều khiển sau khi nhận và xử lý các tín hiệu từ bên ngoài. Sau khi nhận thông tin, bộ phận điều khiển sẽ phát thông tin để động cơ dịch chuyển theo yêu cầu, hình thành một chuỗi động học. Chức năng của hệ thống điều khiển cũng được phân cấp từ đơn giản như đến chức năng phức tạp.

3. Hệ thống quản lý và vận hành

Đây là phần mềm được cài đặt trên máy tính, noi các kỹ thuật viên lập trình các thao tác của robot. Hệ thống quản lý và điều khiển sẽ được kết nối với nhau để truyền và nhận tin.

1.4.3. CÁC ỨNG DỤNG CỦA VR TRONG ĐIỀU KHIỂN CÁNH TAY ROBOT

Trong các nhà máy thông minh, cánh tay robot được ứng dụng cho nhiều mục đích khác nhau. Với tốc độ cao, độ chính xác lớn, độ rung thấp, cánh tay robot không chỉ có thể cải thiện đáng kể hoạt động sản xuất, giúp lắp ráp chính xác cao mà còn hoạt động ổn định khi nó di chuyển nhanh chóng đến một địa điểm cụ thể.

Bên cạnh đó, cánh tay robot cũng có thể được tùy chỉnh để làm các tác vụ hỗ trợ định lượng trong các dây chuyền sản xuất và góp phần nâng cao năng suất nhờ các báo cáo dữ liệu được trích xuất trong thời gian thực, thông qua tích hợp cảm biến, thuật toán tự học, đo lường và tích hợp với hệ thống ERP, MES trong sản xuất.

Một vài ứng dụng phổ biến của cánh tay robot phổ biến như sau:

1.4.3.1. XẾP CHỒNG HÀNG HÓA LÊN CÁC PALLET - BLOCK STACKING

Đây được gọi là dòng Robot Palletizing hay là Block Stacking. Các thuật toán lập trình cho phép dòng robot này có thể gấp hàng từ ray trượt rồi xếp chồng và lắp đầy vào các pallet một cách tuần tự. Với trọng tải lớn, nhiều tay máy robot có khả năng di chuyển các vật liệu nặng như sắt, thép, gỗ, giúp con người loại bỏ nguy cơ gây thương tích.

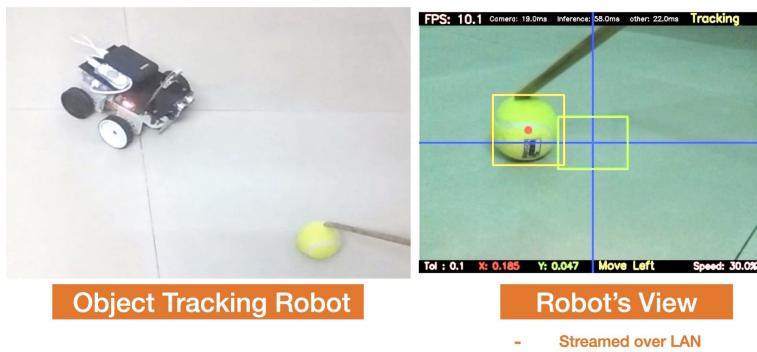
1



Hình 1.4: Cánh tay robot xếp chồng hàng hóa [7]

1.4.3.2. TÌM KIẾM VẬT THỂ ĐỐI TƯỢNG - OBJECT TRACKING

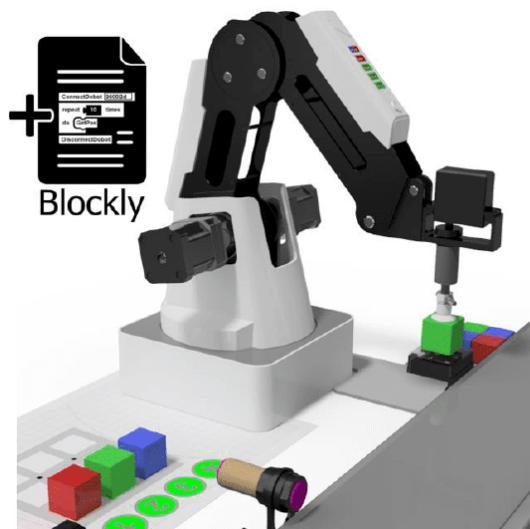
Trên cánh tay robot ngoài thực tế sẽ được tích hợp 1 camera gắn cạnh bên để có thể nhận diện và xác định vật thể để thực hiện các khôi chúc năng theo yêu cầu của người dùng. Mục đích của việc này là nâng cao hiệu suất làm việc, giảm bớt các thao tác thủ công về mặt di chuyển cũng như giảm bớt thời gian nghỉ trong việc lựa chọn vật phẩm, đối tượng.



Hình 1.5: Cánh tay robot xác định vật thể, đối tượng [3]

1.4.3.3. SẮP XẾP CÁC KHỐI MÀU - COLOR SORTING

Vì đã được hỗ trợ tính năng Object Tracking, dựa vào đó cánh tay Robot của chúng ta có thể thực hiện chức năng nhận diện các khối màu và thực hiện phân loại khối màu đến các vị trí nhất định. Ứng dụng của điều này để phân loại các sản phẩm hàng hóa theo màu của chúng đến từng vị trí mà người điều khiển xác định.



Hình 1.6: Cánh tay robot sắp xếp các vật thể theo màu [1]

1.4.3.4. GẮP, THẢ SẢN PHẨM VÀ PHÂN LOẠI HÀNG HÓA - CLASSIFICATION

Robot gấp thả (Pick & Place Robot arms) thường được trang bị hệ thống nhận diện để xác định đồ vật, sau đó tự động nhặt đồ lên và đặt chúng lên bề mặt theo một vị trí và hướng đã định. Hoạt động này thường được sử dụng để phân loại sản phẩm, lắp ráp, cho vào bao bì để đóng gói, ... một cách tự động, giúp gia tăng tốc độ sản xuất và phân phối hàng hóa.



Hình 1.7: Cánh tay robot gấp và thả hàng hóa [7]

1.5. CÔNG NGHỆ VR/AR

1.5.1. KHÁI NIỆM CÔNG NGHỆ VR

VR (**V**irtual **R**eality, tạm dịch: thực tế ảo) là một công nghệ hiện đại được tích hợp cho các thiết bị công nghệ, điện tử. Ví dụ điển hình của công nghệ VR là: Google Cardboard, kính thông minh Samsung Gear VR, ứng dụng thiết kế website 3D cho nhiều lĩnh vực,...



Hình 1.8: Ảnh mô tả về thực tế ảo

Công nghệ VR đưa người dùng bước vào một không gian mô phỏng nhưng vẫn rất chân thực chỉ bằng chiếc kính 3 chiều (kinh thực tế ảo). Thế giới ảo mà người dùng nhìn thấy thực chất được thiết lập và điều khiển bởi một hệ thống máy tính có cấu hình cao.

Phản hồi dựa trên thời gian thực là ưu điểm lớn nhất của công nghệ VR. Tính năng này giúp môi trường 3D có khả năng biến đổi để phù hợp với thế giới thực. Khi người dùng hoạt động, cảm biến của kính 3 chiều sẽ thu nhận tín hiệu điện và truyền đến máy tính để phân tích, sau đó tạo ra một không gian ảo mới theo một nguyên lý hoặc thuật toán sẵn có.

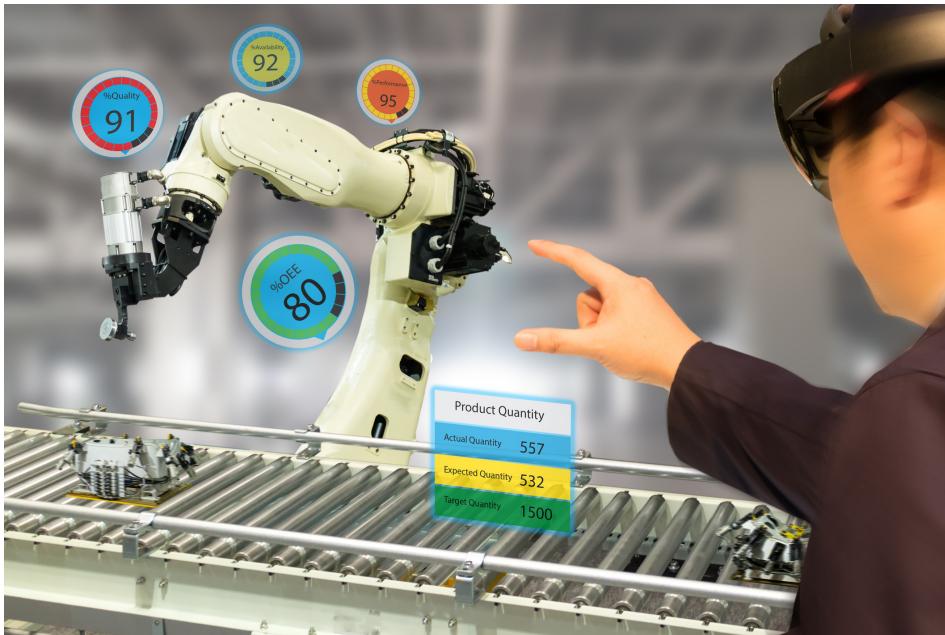
Ví dụ: Khi chơi game nhập vai VR, bạn chỉ cần xoay sang hướng khác thì khung cảnh sẽ lập tức thay đổi theo.

1.5.2. KHÁI NIỆM CÔNG NGHỆ AR

AR (**Augmented Reality**, tạm dịch: thực tế ảo tăng cường) là công nghệ mới được tăng cường từ công nghệ VR. Công nghệ này có khả năng xóa bỏ ranh giới giữa thế giới thực và mô hình 3D ảo. Nghĩa là người dùng sẽ được trải nghiệm mô hình ảo trong không gian thực tế thông smartphone hoặc máy tính.

Những thành phần ảo của công nghệ AR được tạo ra từ máy tính và tái hiện lại ngay trong không gian bạn đang ở. Chúng không chỉ xuất hiện trong môi trường thực, mà còn được bổ trợ thêm âm thanh, video, và đồ họa, tạo ra một trải nghiệm chân thật và sinh động nhất. Với tính năng này, người dùng có thể tương tác hoàn toàn với các mô hình ảo qua điện thoại, bao gồm việc chạm vào, cầm lấy, và đẩy mạnh chúng. Một minh chứng điển hình và phổ biến nhất của công nghệ AR là trò chơi Pokemon Go hoặc với việc sử dụng AR trong môi trường nhà máy robot để cải thiện quy trình sản xuất và tối ưu hóa hoạt động của các robot trong nhà máy.

aar



Hình 1.9: Ảnh mô tả về thực tế ảo tăng cường [32]

1.5.3. CÁC ỨNG DỤNG TRONG CÔNG NGHỆ VR/AR

1.5.4. ỨNG DỤNG TRONG LĨNH VỰC ROBOT

Trong lĩnh vực robot, việc sử dụng công nghệ VR cung cấp môi trường an toàn cho việc huấn luyện người điều khiển robot. Đối với đề tài này, nhóm chúng tôi có thể cung cấp một giải pháp trong đó người dùng có khả năng điều khiển robot thật và quan sát trạng thái của robot thông qua một mô hình 3D ánh xạ chuyển động, tạo ra một trải nghiệm chân thực và an toàn trong môi trường ảo.

1.5.4.1. ỨNG DỤNG CỦA THỰC TẾ ẢO TRONG QUẢNG CÁO VÀ MARKETING

Bên cạnh Wensite 3D, chụp hình 360 độ,... thì việc sử dụng VR trong tiếp thị kỹ thuật số giúp nâng cao trải nghiệm mua sắm của khách hàng bằng cách tăng tương tác trực tiếp, chạm vào và dùng thử sản phẩm mà họ muốn mua thông qua các hiệu ứng hình ảnh đa chiều do VR tạo ra. Ví dụ: trong thời trang trực tuyến, việc lựa chọn sản phẩm có kích thước



Hình 1.10: Sử dụng VR trong lĩnh vực robot [8]

và kiểu dáng phù hợp sẽ được sắp xếp hợp lý hơn khi khách hàng trải nghiệm tính năng VR cho phép họ dùng thử sản phẩm. Ngoài ra, việc lắp



Hình 1.11: Sử dụng VR trong chiến dịch quảng cáo và marketing [39]

đặt các màn hình thực tế tăng cường (AR) ở những nơi công cộng chắc chắn là một chiến lược marketing tiềm năng. Thông thường các thương hiệu lớn như Pepsi, Coca Cola, Axe hay National Geographic đều sử dụng cách này để cải thiện nâng cao giá trị thương hiệu.

1

1.5.4.2. ỨNG DỤNG CỦA THỰC TẾ ẢO KHI XEM PHIM GIẢI TRÍ

Hình 1.12: Sử dụng VR trong việc xem phim giải trí [39]

Khi bạn sử dụng kính thực tế ảo VR để xem phim trên màn hình tivi hay thiết bị máy tính sẽ mang đến cho bạn những trải nghiệm vô cùng tuyệt vời như đang xem phim tại rạp. Netflix đã tự áp dụng VR, chế độ xem VR của thương hiệu này tương thích với kính Gear VR Samsung. Có thể nói, sở hữu sản phẩm tai nghe VR mang thương hiệu Samsung mang đến cho người dùng nhiều trải nghiệm nhất về công nghệ VR.

1.5.4.3. ỨNG DỤNG CỦA THỰC TẾ ẢO TRONG NGÀNH DU LỊCH ẢO

Nhiều công ty đã tung ra các sản phẩm du lịch ảo với giá cả phải chăng nhất. Chỉ bằng cách đeo kính VR, bạn có thể trải nghiệm đi đến điểm du lịch thực tế mà không cần quá tốn kém nhiều chi phí. Trong thế giới ảo, bạn sẽ như lạc vào những viện bảo tàng, di tích văn hóa, hòa mình vào thế giới tự nhiên, chiêm ngưỡng những danh lam thắng cảnh tuyệt đẹp của thế giới cũng như tại Việt Nam.



Hình 1.13: Sử dụng VR trong ngành du lịch [39]

1.5.4.4. ỨNG DỤNG THỰC TẾ ẢO TRONG ĐÀO TẠO BÁC SĨ

Ứng dụng mô phỏng trong môi trường thực tế ảo mang tên Human-Sim giúp đào tạo các chuyên gia y tế bằng cách đưa họ vào các tình huống hiểm nghèo, như phải mô phỏng lại một trường hợp bệnh nhân bị chấn thương nặng và họ phải xử trí thành công vấn đề [39]. Mặc dù nó không thể tái tạo mức độ căng thẳng của một phương pháp điều trị thực tế, nhưng HumanSim làm cho các bác sĩ hành nghề hứng thú và say mê hơn với công việc của họ bằng cách mô phỏng chi tiết các quy trình khó khăn mà họ phải đối mặt trong thế giới thực. [39]

1.5.4.5. ỨNG DỤNG THỰC TẾ ẢO TRONG THIẾT KẾ XE HƠI

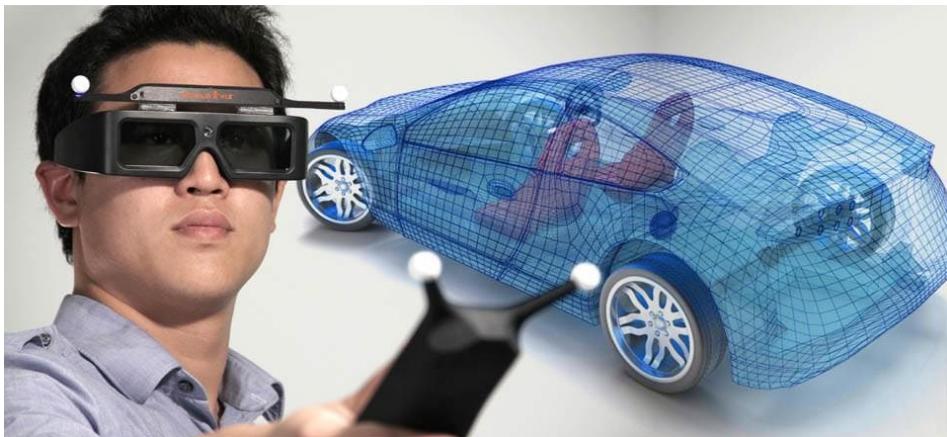
Nhiều nhà sản xuất ô tô như Ford từ lâu đã sử dụng các phòng thí nghiệm tái hiện bằng thực tế ảo để cho phép các nhân viên trải nghiệm một chiếc xe trước khi nó được chế tạo. Từ đây, họ có thể xem chi tiết các bộ phận của từng chiếc xe và thậm chí cho phép người dùng đánh giá chúng và sau đó nhận phản hồi trước khi mô hình được chế tạo.

Điều này cho phép Ford cập nhật các thiết kế của mình nhanh hơn bao giờ hết để phù hợp với xu thế chung. Range Rover cũng sử dụng công nghệ thực tế ảo có tên CAVE (Computer-Aided Virtual Environment) trong

1



Hình 1.14: Sử dụng VR trong quá trình đào tạo bác sĩ [39]

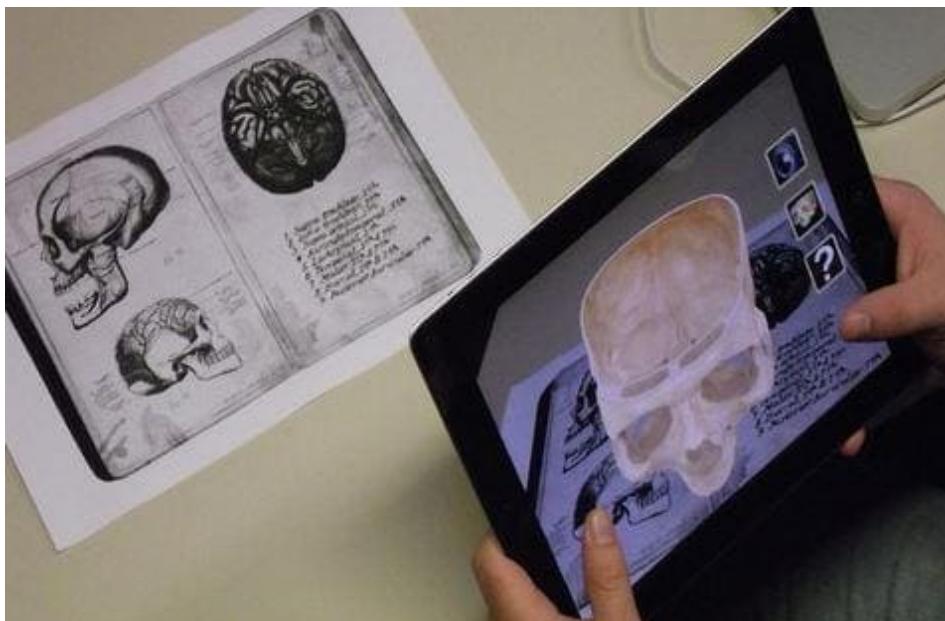


Hình 1.15: Sử dụng VR trong quá trình thiết kế xe hơi [39]

quá trình thiết kế, cho phép người dùng nhìn thấy và thậm chí tương tác với các đối tượng trước khi sản xuất. Cấu hình bên trong, không gian cabin, khí động học và hành vi va chạm có thể được kiểm tra trong thực tế ảo mà không cần tốn thời gian và tiền bạc vào việc chế tạo mô hình. [39]

1.5.4.6. ỨNG DỤNG THỰC TẾ ẢO TRONG NGÀNH GIÁO DỤC

Giáo dục cũng là một ứng dụng vô cùng tiềm năng khi ứng dụng công nghệ thực tế ảo, nó cho phép các học sinh được tham quan các công trình kiến trúc cổ đại trong giờ lịch sử, tham gia các bài tập giải phẫu trong lớp sinh học hoặc được truyền cảm hứng từ những hiện tượng thiên nhiên kỳ thú mà sách vở có thể không bao giờ mô tả được.



Hình 1.16: Sử dụng VR trong giáo dục [39]

1.6. GIỚI THIỆU VỀ ĐỒ ÁN

Việc áp dụng ứng dụng VR để điều khiển cánh tay Robot trên môi trường thực tế ảo có những thách thức mà ta phải đối mặt đó là:

- Độ chân thực và hiệu suất tương tác: Việc tạo ra một môi trường ảo chân thực và đảm bảo hiệu suất tương tác giữa người dùng và cánh tay robot là một thách thức. Các yếu tố như độ trễ, độ nhạy, và độ chính xác của hệ thống VR cần được cải thiện để đảm bảo trải nghiệm người dùng tốt nhất.

1

- Bảo mật và an toàn: Trong quá trình tương tác với cánh tay robot trong môi trường ảo, ta còn cần đặc biệt chú ý đến các vấn đề liên quan đến bảo mật dữ liệu và an toàn của người dùng. Việc bảo vệ khỏi việc tấn công mạng và đảm bảo rằng cánh tay robot không gây nguy hiểm trong thế giới thực là cực kì quan trọng.
- Phát triển và tích hợp phần mềm: Việc phát triển phần mềm ứng dụng VR và tích hợp nó với cánh tay Robot đòi hỏi kiến thức sâu rộng về cả hai lĩnh vực. Sự đồng bộ và tương thích giữa phần mềm và phần cứng là một thách thức kỹ thuật.
- Hạn chế về thiết bị: Các thiết bị VR hiện nay có thể đặt ra hạn chế về khả năng xử lý và hiển thị đồ họa. Điều này có thể ảnh hưởng đến khả năng tái tạo một môi trường ảo phức tạp và hiển thị chi tiết, đặc biệt khi liên quan đến cánh tay robot có nhiều bậc tự do.
- Đào tạo và chấp nhận từ người sử dụng: Người sử dụng cần được đào tạo để hiểu cách tương tác với cánh tay robot trong không gian ảo. Sự chấp nhận và thoái mái khi sử dụng công nghệ VR cũng là một thách thức, đặc biệt là trong môi trường công nghiệp có nguy cơ rủi ro cao. Để triển khai chương trình đào tạo này tại một nhà máy trong khu công nghệ cao TP.HCM, với mục tiêu giúp học viên làm quen với module cánh tay robot trước khi thực hiện thao tác thực tế với thiết bị thực tế trong nhà máy. Điều này giúp tăng tốc quá trình đào tạo và giảm thiểu rủi ro và hỏng hóc khi làm việc với thiết bị thực tế. [29]
- Chi phí đầu tư: Việc triển khai hệ thống VR và cảm biến để điều khiển cánh tay robot có thể đòi hỏi chi phí đầu tư lớn. Điều này bao gồm cả việc mua sắm thiết bị VR chất lượng cao và phát triển phần mềm tương ứng.
- Hiệu suất thực tế: Hiệu suất của cánh tay robot khi được điều khiển thông qua VR cần được đánh giá và so sánh chặt chẽ với việc điều khiển truyền thống. Việc này sẽ đặt ra các thách thức về độ chính xác và độ tin cậy của hệ thống trong môi trường công nghiệp thực tế.
- Nghiên cứu về sự tương tác và tiện ích thực tế: Nghiên cứu cần tập trung vào việc hiểu rõ hơn về cách người dùng tương tác với cánh tay robot trong môi trường ảo và cách nó có thể đóng góp vào tiện ích thực tế trong quá trình sản xuất. Điều này đòi hỏi nghiên cứu về nhận thức, phản ứng, và trải nghiệm người dùng.

Vì những khó khăn thách thức trên, nhóm đã quyết định thực hiện nghiên cứu việc ứng dụng VR để điều khiển cánh tay Robot trong công nghiệp nhằm đưa ra những kết luận và rủi ro về tính khả thi của việc hiện thực điều khiển cánh tay Robot trong ứng dụng thực tế ảo. Với mong muốn có thể tạo ra một bản sao ảo của cánh tay robot trên ứng dụng VR nhóm cũng muốn đánh giá được tính khả thi về mặt hiện thư, cũng như về hiệu năng và đánh giá khách quan về việc ứng dụng trong đời sống của việc hiện thực này.

1.7. BỘ CỤC CỦA ĐỒ ÁN

Phần tiếp theo của Đồ án sẽ được chia thành 4 chương: Ở chương 2 sẽ giới thiệu tổng quan về các loại giao thức M2M (MQTT, RTSP và HTTP) cũng như các công nghệ được nghiên cứu và sử dụng trong Đồ án. Chương 3 sẽ tiến hành thiết kế kiến trúc hệ thống và hiện thực điều khiển cánh tay Robot. Chương 4 là phần đánh giá kết quả Đồ án mà nhóm đã làm được. Và cuối cùng là chương 5, tổng kết và đề ra những định hướng có thể phát triển trong tương lai.

2

CƠ SỞ LÝ THUYẾT

Chương cơ sở lý thuyết cung cấp những khái niệm nền tảng cho dự án, bao gồm khái niệm về Digital Twin, tổng quan về các giao thức M2M đang được dùng phổ biến hiện nay trong kết nối giữa máy và máy cùng với một số khái niệm về toạ độ trong không gian 3D của môi trường thực tế ảo (VR), đặc biệt là trong ngữ cảnh của Unity. Chương sẽ giới thiệu qua hai khái niệm quan trọng là toạ độ Euler và Quaternion, giúp mô tả vị trí và hướng của các đối tượng trong không gian 3D một cách chính xác và linh hoạt. Cuối cùng việc hiểu rõ về toạ độ hướng và góc nghiêng của camera, được sử dụng để lấy hình ảnh trực tiếp từ kính VR, sẽ đóng vai trò quan trọng trong quá trình tương tác giữa môi trường ảo và thực. Đây là những kiến thức cơ bản nhưng quan trọng, giúp định rõ vị trí và hướng của người dùng trong không gian ảo, tạo nền tảng cho việc triển khai các tính năng và chức năng phức tạp hơn trong dự án.

2.1. DIGITAL TWIN

Công nghệ digital twin (Bản sao kỹ thuật số) là một khái niệm mới đang nổi lên, dần trở thành trung tâm của sự chú ý trong nền công nghiệp và dạo gần đây là trong các nghiên cứu. Sự phát triển của nền công nghiệp 4.0 và các công nghệ thời đại mới đã và đang tạo những bước phát triển lớn đối với nền công nghiệp thế giới. Internet of Things (IoT - mạng kết nối vạn vật) đã làm tăng lên số lượng dữ liệu được trao đổi trong hoạt động sản xuất, chăm sóc sức khỏe hay thành phố thông minh, từ đó tạo ra nhu cầu phân tích dữ liệu nhằm đưa ra những đánh giá, dự đoán, xác định lỗi, mô phỏng... Digital Twin xuất hiện như một giải pháp cho vấn đề tích hợp liền mạch giữa IoT và phân tích dữ liệu qua việc tạo ra sự kết nối giữa thực thể vật lý và bản sao ảo. Mỗi trường Digital Twin cho phép thực hiện các phân tích nhanh chóng và đưa ra quyết định thời gian thực thông qua những phân tích chính xác. Ở phần tiếp theo, ta sẽ tìm hiểu định nghĩa cụ thể của Digital Twin trong các giai đoạn lịch sử, và giải đáp những hiểu lầm thường thấy về Digital Twin. Tiếp đến, ta nhìn nhận những ứng dụng của Digital Twin trong thời đại công nghiệp 4.0, từ đó xác định những ưu, nhược điểm và thử thách của công nghệ này đối với sự phát triển của thời đại.

2

2.1.1. KHÁI NIỆM VỀ DIGITAL TWIN

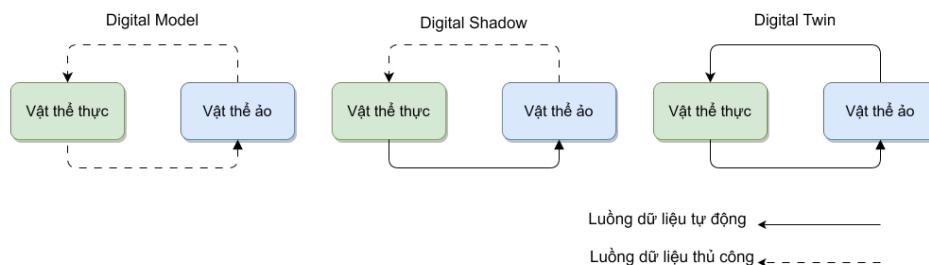
Khái niệm về Digital Twin đã xuất hiện từ khoảng những năm 2000. Cho đến nay, đã có những định nghĩa khác nhau về digital twin. Năm 2012, NASA [16] cho rằng "Digital Twin là một mô phỏng tích hợp đa vật lý, đa cấp độ, mang tính xác suất (multiphysics, multiscale, probabilistic) của một phương tiện hay hệ thống đã tồn tại sử dụng mô hình tái hiện vật lý, cập nhật cảm biến, lịch sử... tốt nhất để ánh xạ sự tồn tại của vật thể đang bay". NASA đã từng hiện thực một digital twin cho tên lửa của họ trong chương trình Apollo, sau đó quân đội không quân Mĩ (US Air Force) cũng theo chân NASA và sử dụng công nghệ Digital Twin cho thiết kế, bảo trì và dự đoán sức khỏe các chiến cơ của họ. [37]. Năm 2017, Chen [10] cho rằng Digital twin là một mô hình vi tính hóa của một thiết bị vật lý hay hệ thống có khả năng thể hiện mọi chức năng và kết nối với các phần tử đang hoạt động". Zheng [48] định nghĩa Digital Twin như "Một tập thông tin ảo có khả năng mô tả tiềm năng hoặc quy trình sản xuất thực tế từ mức độ nguyên tử micro (micro atomic level) tới cấp độ hình học vĩ mô (macro geometrical level). Định nghĩa khác của của Madni [27]

2

(2019) rằng "Digital Twin là một sự tồn tại ảo của một hệ thống vật lý luôn luôn được cập nhật với trạng thái, sự bảo trì và sức khỏe của hệ thống vật lý trong suốt vòng đời của nó." Theo nhóm, dù có nhiều định nghĩa khác nhau, nhưng chúng vẫn có chung các ý chính về Digital Twin là một bản sao ảo của vật thể ở thế giới thực, và bản sao này sẽ luôn luôn trao đổi, thu thập dữ liệu của vật thể thực nhằm phản ánh chân thực nhất trạng thái, sức khỏe của vật thể. Kèm theo đó, với dữ liệu thu thập được từ trạng thái của vật thể thật qua thể ảo, ta có thể sử dụng nó để đánh giá, phân tích, mô phỏng và đưa ra những quyết định phù hợp.

2.1.2. NHỮNG NHẦM LÃN VỀ DIGITAL TWIN

Digital Twin là một thực thể ảo trong môi trường ảo, phản ánh trạng thái của thiết bị thật và mang tính cá nhân hóa cao, nhưng điều đó không có nghĩa rằng chỉ cần tạo ra một mô hình ảo giống với thực tế thì là digital twin. Để làm rõ hơn về digital twin, Aidan Fuller [15] đã phân tách những khái niệm thường bị nhầm lẫn khi nói về digital twin như sau:



Hình 2.1: Một số khái niệm nhầm lẫn của Digital Twin

- **Mô hình ảo (Digital Model):** Một mô hình ảo là một phiên bản số của một vật thể thực đã tồn tại hoặc dự định phát triển. Đối với một mô hình ảo, không hề có luồng dữ liệu tự động trao đổi giữa vật thể thực và mô hình ảo, một ví dụ thường thấy là mô hình ảo của tòa nhà, mô hình thiết kế hay phát triển sản phẩm. Điều này có nghĩa là thay đổi của vật thể thực sẽ không ảnh hưởng đến mô hình ảo, và ngược lại.
- **Bóng ma ảo (Digital Shadow):** Bóng ma ảo bao gồm mô hình ảo của vật thể thực, nhưng lần này có thêm luồng dữ liệu từ vật thể

thực sang mô hình ảo. Sự thay đổi trạng thái của vật thể thực sẽ ảnh hưởng tới mô hình ảo, nhưng không có chiều ngược lại.

- **Bản sao ảo (Digital Twin):** Một mô hình ảo được gọi là digital twin khi có sự luân chuyển 2 chiều của dữ liệu, từ vật thể thực sang mô hình ảo và ngược lại. Điều này có nghĩa là thay đổi của vật thể thực cũng ảnh hưởng tới mô hình ảo, và thông qua mô hình ảo, ta có thể thay đổi, tác động lên vật thể thực.

2

2.1.3. ỨNG DỤNG CỦA DIGITAL TWIN

Digital Twin đã và đang được vận dụng trong nhiều lĩnh vực khác nhau. Khái niệm và định nghĩa của Digital Twin đang dần phát triển trong môi trường học thuật, cùng với sự cải tiến không ngừng của các công nghệ IoT và AI. Hiện nay, ứng dụng được quan tâm chủ yếu của digital twin thường được thấy là các thành phố thông minh, sản xuất và một số ứng dụng liên quan tới sức khỏe [15]

2.1.3.1. DIGITAL TWIN TRONG THÀNH PHỐ THÔNG MINH

Việc sử dụng và tiềm năng cho Digital Twin phát triển mạnh mẽ trong lĩnh vực thành phố thông minh đang tăng lên hằng năm nhờ vào sự phát triển nhanh của các kết nối thiết bị IoT. Với một lượng lớn các thành phố thông minh được xây dựng, sự kết nối giữa các cộng đồng càng cao, thì nhu cầu của digital twin cũng tăng cao. Lượng dữ liệu thu thập được từ các cảm biến IoT đang tích hợp trong các khu vực nhà thông minh cũng dần mở rộng, tạo đường cho những nghiên cứu nhằm tối đa hóa những ứng dụng AI nâng cao.

Các dịch vụ và kiến trúc hạ tầng trong thành phố thông minh hoàn toàn có thể được cải thiện trong tương lai thông qua dữ liệu thu thập từ cảm biến được rải rác trong khu vực này. Dữ liệu có thể được sử dụng cho hỗ trợ việc thiết kế và phát triển các thành phố thông minh mới. Ngoài lợi ích đem lại trong việc thiết kế, dữ liệu còn giúp ích trong việc tiết kiệm năng lượng sử dụng. Chính sự phát triển của nhà thông minh, sự tăng lên của thiết bị IoT và dữ liệu thu thập đã và đang tạo ra tiềm năng ứng dụng công nghệ Digital Twin. Digital Twin giúp cải thiện tình trạng thành phố bằng cách tạo các mô phỏng sống trong môi trường ảo, thử nghiệm các kịch bản khác nhau, và đồng thời bản sao ảo của thành phố cũng học

được từ môi trường thực tế bằng việc phân tích và giám sát dữ liệu nhận được.

2

2.1.3.2. DIGITAL TWIN TRONG SẢN XUẤT

Ứng dụng nổi bật tiếp theo của Digital Twin là trong môi trường sản xuất công nghiệp. Lý do chính cho việc này là do nhà sản xuất luôn cố gắng tìm cách kiểm soát và theo dõi việc sản xuất sản phẩm nhằm tìm giải pháp tiết kiệm thời gian và tiền bạc, là động lực và mục đích chính của mọi nhà sản xuất. Chính vì vậy, Digital Twin trở thành một lựa chọn hấp dẫn để ứng dụng trong môi trường sản xuất. Digital Twin mang lại khả năng cung cấp trạng thái thời gian thực về hiệu năng của máy móc và phản hồi của dây chuyền sản xuất, qua đó giúp cho nhà sản xuất tận dụng dữ liệu cho các dự đoán, cải thiện và phát hiện vấn đề sớm hơn.

Digital twin tăng cường kết nối và phản hồi giữa các thiết bị, qua đó, cải thiện độ tin cậy và hiệu năng. Digital Twin kết hợp với AI giúp tăng cường độ chính xác, khi digital twin không ngừng thu thập và lưu trữ lượng dữ liệu khổng lồ, từ đó được sử dụng bởi AI cho phân tích, dự đoán hiệu năng. Digital Twin cũng có thể tạo ra một môi trường để kiểm thử sản phẩm, hay một hệ thống hoạt động dựa trên dữ liệu thực tế thay vì dữ liệu giả định, giúp cho nhà sản xuất đưa ra những đánh giá, hành động hiệu quả và thực tế.

Một ứng dụng khác của digital twin được thực hiện hiện nay là trong ngành sản xuất ô tô, mà cụ thể bởi Tesla. Khả năng hiện thực một Digital Twin của bộ phận xe hơi trở nên rất giá trị vì có thể sử dụng bản sao ảo đó cho mô phỏng và phân tích dữ liệu. AI được sử dụng để phân tích dữ liệu thực tế trên xe, qua đó dự đoán trạng thái hiện tại và tương lai của các bộ phận xe.

Ngành xây dựng cũng là một ngành thu được lợi ích từ việc ứng dụng Digital Twin. Theo dõi giai đoạn của công trình xây dựng hay thiết kế cấu trúc là những ứng dụng tiềm năng. Với việc theo dõi chất lượng công trình thật thông qua bản sao ảo, ta có thể biết được những thay đổi, ảnh hưởng của cấu trúc công trình, nhận biết sớm những vấn đề từ đó đưa ra bảo trì và sửa chữa nhanh chóng. Digital Twin cũng có thể được tạo ra trước khi xây dựng công trình, sử dụng những dữ liệu thực tế từ các công trình trước để thử nghiệm các mô phỏng

Điểm chung từ việc sử dụng digital twin trong các lĩnh vực trên là nhằm lợi dụng khả năng hiện thực mô phỏng thời gian thực sử dụng dữ liệu thực tế thay vì những model tĩnh thiếu chi tiết. Các model tĩnh cũng có tầm quan trọng của chúng, tuy nhiên thiếu sót các thông số thời gian thực và chi tiết sẽ làm giới hạn khả năng dự đoán và học hỏi. Digital Twin có thể học từ dữ liệu thực tế, dùng để quan sát trạng thái của vật thể thực, hay áp dụng lên nó những giải thuật học máy, học sâu.

2.1.3.3. DIGITAL TWIN TRONG CHĂM SÓC SỨC KHỎE

Ngành chăm sóc sức khỏe là một lĩnh vực tiềm năng cho ứng dụng Digital Twin. Một ứng dụng tương lai có thể phát triển là Digital twin của cơ thể con người, đưa ra được dữ liệu thời gian thực của cơ thể. Ứng dụng thực tế hơn đang dùng hiện nay là sử dụng digital twin nhằm mô phỏng ảnh hưởng của một số loại thuốc, hay lập kế hoạch và thực hiện phẫu thuật.

Digital Twin trong chăm sóc sức khỏe đem lại khả năng giúp cho nhà nghiên cứu, bác sĩ, bệnh viện và dịch vụ chăm sóc sức khỏe mô phỏng môi trường cần thiết. Một lần nữa, Digital Twin có thể kết hợp với AI để đưa ra những dự đoán và quyết định chính xác. Sự phát triển của Digital Twin trong ngành này vẫn còn non yếu, tuy nhiên tiềm năng phát triển của nó vẫn rất cao, như việc quản lý giường bệnh tới quản lý bệnh viện.

Có thể thấy, Digital Twin vẫn là một ứng dụng tiềm năng cho thời đại công nghiệp 4.0, nhất là khi sự phát triển của IoT và AI đang có những bước tiến vượt bậc. Digital twin dự đoán sẽ đem lại nhiều đổi mới cho sản xuất và cách thức vận hành của doanh nghiệp. Ở phần đồ án này, nhóm sẽ tập trung nghiên cứu và ứng dụng một trong những khía cạnh quan trọng của Digital Twin, đó chính là giao thức hỗ trợ truyền tải dữ liệu sao cho đạt được khả năng truyền tải dữ liệu thời gian thực, nhanh và đáng tin cậy. Phần tiếp theo sẽ nói về các giao thức M2M (machine to machine) phổ biến đang được sử dụng hiện nay, cụ thể là hai giao thức RTSP, MQTT và HTTP.

2.2. TỔNG QUAN VỀ CÁC GIAO THỨC M2M

Giao tiếp Machine-to-Machine (M2M) là một công nghệ đầy hứa hẹn cho các hệ thống giao tiếp thế hệ tiếp theo. Mô hình giao tiếp này tạo điều kiện cho các giao tiếp phổ biến với cơ chế tự động hóa hoàn toàn, trong đó một số lượng lớn thiết bị thông minh được kết nối bằng liên kết có dây/không dây, tương tác với nhau mà không cần sự can thiệp trực tiếp của con người. Do đó, giao tiếp M2M thường được ứng dụng trong các lĩnh vực rộng lớn như lưới điện thông minh, chăm sóc sức khỏe điện tử, mạng khu vực gia đình, hệ thống giao thông thông minh, giám sát môi trường, thành phố thông minh và tự động hóa công nghiệp. Để hiểu rõ hơn về công nghệ này, chúng ta sẽ cùng nhau tìm hiểu về Khái niệm, Cách thức hoạt động và Ứng dụng của nó trong đời sống.

2.2.1. GIỚI THIỆU VỀ M2M

M2M là viết tắt của “Machine to Machine.” Đó là một thuật ngữ mô tả bất kỳ công nghệ nào cho phép các thiết bị có kết nối mạng trao đổi thông tin và thực hiện các hành động mà không cần sự trợ giúp thủ công của con người. Nói cách khác, giao tiếp được thực hiện từ máy này sang máy khác.

Công nghệ M2M cho phép các thiết bị trên mạng đưa ra quyết định tự chủ mà không yêu cầu các thao tác thủ công. Mặc dù được sử dụng rộng rãi trong sản xuất, nhưng nó cũng được sử dụng trong các lĩnh vực khác. Các ứng dụng phổ biến bao gồm chăm sóc sức khỏe, bảo hiểm và Internet vạn vật (IoT).

Khi ở trong ngành sản xuất, nó thường được kết hợp với các công nghệ khác như SCADA. Một ví dụ về kết nối M2M trong các lĩnh vực khác là máy bán hàng tự động. Nó tự động gửi thông tin kiểm kê hàng trong kho của mình cho người điều phối. Một ví dụ điển hình khác là máy ATM tự gửi thông tin khi gần hết tiền mặt cho cơ quan chức năng của nó.

2.2.2. NGUYÊN LÝ HOẠT ĐỘNG CỦA M2M

Kết nối M2M có vẻ phức tạp, nhưng chúng hoạt động theo nguyên tắc đơn giản. Chúng lấy dữ liệu cảm biến tự động và chuyển dữ liệu đó

qua mạng truy cập công cộng (public access network). Không giống như SCADA, các kết nối M2M sử dụng các mạng công cộng như mạng di động hoặc Ethernet. Đây là những gì làm cho các kết nối M2M trở nên hiệu quả về chi phí.

2

Một hệ thống M2M điển hình bao gồm nhiều cảm biến, RFID, Wi-Fi hoặc liên kết truyền thông di động (cellular communications link). Các cảm biến giao tiếp một số key condition với máy tính. Chẳng hạn, trong máy ATM, một cảm biến phát hiện mức tiền sẽ gửi tín hiệu khi tiền ở mức thấp.

Các hệ thống M2M cũng thường chứa phần mềm tính toán tự trị (autonomic computing software). Phần mềm này được thiết kế để giúp mạng diễn giải dữ liệu mà nó nhận được và hành động tương ứng.

Ngoài khả năng giám sát thiết bị và hệ thống từ xa, những lợi ích hàng đầu của M2M bao gồm:

- **Giảm chi phí** bằng cách giảm thiểu thời gian bảo trì và ngừng hoạt động của thiết bị;
- **Tăng doanh thu** bằng cách mở ra các cơ hội kinh doanh mới để phục vụ các sản phẩm trong lĩnh vực này; và
- **Cải thiện dịch vụ khách hàng** bằng cách chủ động theo dõi và bảo dưỡng thiết bị trước khi thiết bị hỏng hoặc khi cần thiết.

2.2.3. ỨNG DỤNG CỦA M2M

Kết nối M2M ở xung quanh chúng ta. Dưới đây là một vài ví dụ điển hình:

Máy bán hàng tự động Một cảm biến phát hiện khi một sản phẩm cụ thể sắp hết trong máy bán hàng tự động. Thông tin này được truyền qua Wi-Fi hoặc mạng di động đến máy tính của người điều phối. Quá trình này hoàn toàn tự động.

Y tế từ xa Trong lĩnh vực y tế từ xa, kết nối M2M có thể sử dụng các cảm biến để theo dõi thông tin quan trọng của bệnh nhân. Thông tin này sau đó có thể được chuyển đến một máy cục bộ gửi thuốc hoặc bác sĩ ở xa.

Hệ thống nhà thông minh Có lẽ không lĩnh vực nào sử dụng kết nối M2M nhiều hơn hệ thống nhà thông minh. Có rất nhiều ví dụ về điều này. Điểm mấu chốt là nó cho phép một số cảm biến nhất định kết nối với các thiết bị khác để tự động hóa ngôi nhà của bạn.

2.2.4. MQTT

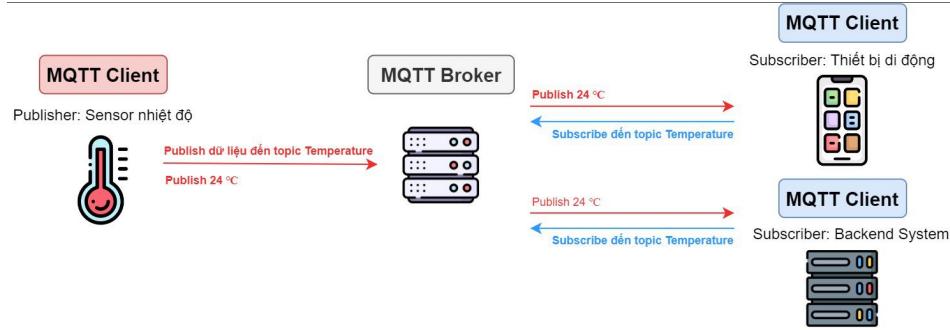
MQTT (Message Queueing Telemetry Transport) được công bố là một giao thức cho các ứng dụng machine-to-machine theo mô hình publish/subscribe cực kì nhẹ và là một giao thức kết nối Internet of Things. Đây là một giao thức truyền thông điệp (message) mở, chủ yếu tập trung vào việc sử dụng băng thông thấp, độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền không ổn định. Do đó, liên lạc giữa các cảm biến thông qua liên kết vệ tinh là một trong những ứng dụng của MQTT. Kể từ năm 2013, MQTT được tiêu chuẩn hóa thành một giao thức cho Internet of Things bởi Tổ chức Organization for the Advancement of Structured Information Standards (OASIS).

Mặc dù MQTT bắt đầu như một giao thức độc quyền được sử dụng để giao tiếp với các hệ thống thu thập dữ liệu và kiểm soát giám sát (SCADA) trong ngành dầu khí, nhưng nó đã trở nên phổ biến trong lĩnh vực thiết bị thông minh và ngày nay là giao thức nguồn mở hàng đầu để kết nối internet vạn vật (IoT) và thiết bị IIoT công nghiệp.

2.2.4.1. NGUYÊN LÝ HOẠT ĐỘNG CỦA MQTT

Nhằm mục đích tối đa hóa băng thông có sẵn, mô hình giao tiếp publish/subscribe (pub/sub) của MQTT là một giải pháp thay thế cho kiến trúc client-server truyền thống - giao tiếp trực tiếp với điểm cuối. Ngược lại, trong mô hình pub/sub, ứng dụng khách gửi tin nhắn (message) (the publisher) được tách biệt khỏi các ứng dụng khách nhận tin nhắn (the subscribers). Bởi vì cả publisher và subscriber đều không liên hệ trực tiếp với nhau, nên bên thứ ba – các broker – sẽ chịu trách nhiệm cho các kết nối giữa chúng.

Ứng dụng khách MQTT (MQTT client) bao gồm publishers và subscribers, thuật ngữ này đề cập đến việc ứng dụng khách đang xuất bản tin nhắn (message) hay đăng ký nhận tin nhắn. Hai chức năng này có thể được triển khai trong cùng một máy khách MQTT. Khi một thiết bị (hoặc



Hình 2.2: Kiến trúc chung của MQTT sử dụng MQTT Broker

máy khách) muốn gửi dữ liệu đến máy chủ (hoặc broker), nó được gọi là *publish*. Khi hoạt động được đảo ngược, nó được gọi là *subscribe*. Theo mô hình pub/sub, nhiều client có thể kết nối với một broker và subscribe các *topic* mà chúng quan tâm.

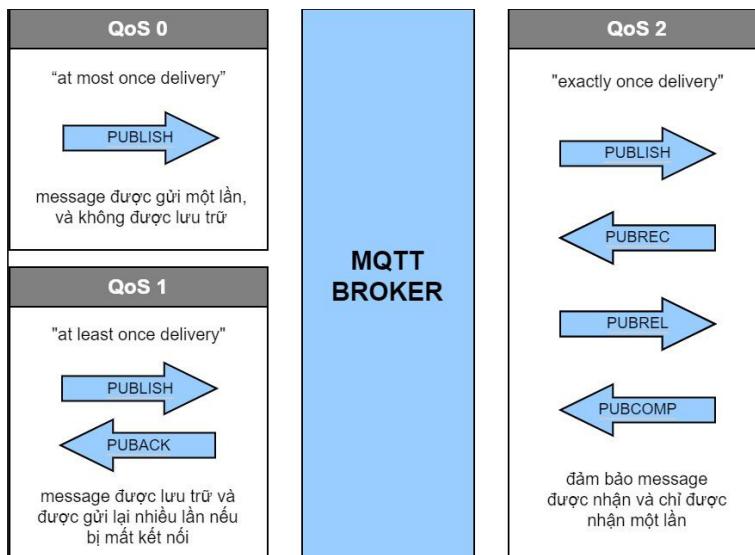
Một bài viết của IBM mô tả mô hình pub/sub: "Những Publisher gửi tin nhắn, những Subscriber nhận tin nhắn mà chúng quan tâm và Broker chuyển tin nhắn từ Publisher đến Subscriber. Publisher và Subscriber là ứng dụng khách MQTT (MQTT client), chỉ giao tiếp với một MQTT broker. Máy khách MQTT có thể là bất kỳ thiết bị hoặc ứng dụng nào (từ bộ vi điều khiển như Arduino đến máy chủ ứng dụng đầy đủ được lưu trữ trên đám mây) chạy thư viện MQTT."

Tin nhắn - Message

Một cách khác để MQTT giảm thiểu quá trình truyền của nó là sử dụng cấu trúc message nhỏ, được xác định chặt chẽ. Mỗi message có một header cố định chỉ 2 byte. Có thể sử dụng header tùy chọn nhưng sẽ làm tăng kích thước của message. Tải trọng message được giới hạn chỉ 256 MB. Ba mức Chất lượng Dịch vụ (QoS) khác nhau cho phép các nhà thiết kế mạng lựa chọn giữa giảm thiểu truyền dữ liệu và tối đa hóa độ tin cậy.

- **QoS 0** – Cung cấp lượng truyền dữ liệu tối thiểu. Với cấp độ này, mỗi message được gửi đến một subscriber một lần mà không cần xác nhận lại. Không có cách nào để biết liệu subscriber có nhận được tin nhắn hay không. Phương pháp này đôi khi được gọi là "fire and forget" hoặc "at most once delivery". Bởi vì cấp độ này giả định rằng

2



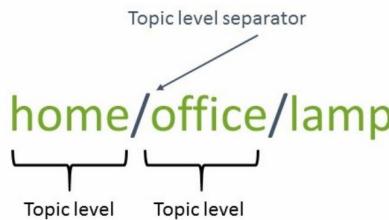
Hình 2.3: Ba mức Chất lượng Dịch vụ (QoS)

quá trình gửi đã hoàn tất, nên các message không được lưu trữ để gửi cho các máy khách đã ngắt kết nối khi nó kết nối lại sau này.

- **QoS 1** – Broker gửi message và sau đó đợi phản hồi xác nhận từ subscriber. Nếu không nhận được xác nhận trong khung thời gian đã chỉ định, message sẽ được gửi lại. Sử dụng phương pháp này, subscriber có thể nhận được tin nhắn nhiều lần nếu broker không nhận được xác nhận của subscriber kịp thời. Và nó đôi khi được gọi là "at least once delivery".
- **QoS 2** – Client và broker sử dụng four-step handshake để đảm bảo rằng message được nhận và message đó chỉ được nhận một lần. Điều này được gọi là "exactly once delivery".

Đối với các trường hợp mà giao tiếp là đáng tin cậy nhưng tài nguyên bị hạn chế, thì QoS 0 có thể là lựa chọn tốt nhất. Đối với các trường hợp mà giao tiếp là không đáng tin cậy, nhưng khi các kết nối không bị giới hạn về tài nguyên, thì QoS 2 sẽ là lựa chọn tốt nhất. Còn QoS 1 là một loại giải pháp tốt cho cả hai trường hợp nhưng nó yêu cầu ứng dụng nhận dữ liệu phải biết cách xử lý các bản sao từ việc gửi message nhiều lần.

Chủ đề - Topic



2

Hình 2.4: Cấu trúc của topic [40]

Message trong MQTT được publish dưới dạng topic. Các Topic là các cấu trúc trong hệ thống phân cấp sử dụng ký tự gạch chéo (/) làm dấu phân cách. Cấu trúc này tương tự như cấu trúc của cây thư mục trên hệ thống tệp máy tính. Một cấu trúc như **sensors/OilandGas/Pressure/** cho phép subscriber chỉ định rằng chỉ nên gửi dữ liệu từ các máy khách publisher lên topic Pressure hoặc để có cái nhìn rộng hơn, có lẽ tất cả dữ liệu từ các máy khách publisher lên bất kỳ topic **sensors/OilandGas** nào . Topic không được tạo rõ ràng trong MQTT. Nếu một broker nhận được dữ liệu được publish cho một topic hiện không tồn tại, topic đó sẽ được tạo đơn giản và client có thể subscribe topic mới.

2.2.4.2. ƯU VÀ NHƯỢC ĐIỂM CỦA MQTT

Ưu điểm

Các thuộc tính lightweight và chi phí tối thiểu của kiến trúc giao thức MQTT giúp đảm bảo truyền dữ liệu trơn tru với băng thông thấp và giảm tải cho CPU và RAM. Các ưu điểm của MQTT so với các giao thức cạnh tranh như sau:

- Truyền dữ liệu hiệu quả và triển khai nhanh chóng do đây là một giao thức lightweight;
- Mức sử dụng mạng thấp, do các gói dữ liệu được giảm thiểu;
- Phân phối dữ liệu hiệu quả;
- Thực hiện thành công remote sensing và điều khiển;
- Truyền tải thông điệp nhanh chóng, hiệu quả;
- Sử dụng lượng điện năng nhỏ, tốt cho các thiết bị được kết nối;

- Tối ưu hóa băng thông mạng.

Nhược điểm

2

Nhược điểm tiềm ẩn đối với MQTT bao gồm:

- MQTT có chu kỳ truyền chậm hơn so với Constrained Application Protocol (CoAP).
- Khám phá tài nguyên của MQTT hoạt động trên đăng ký chủ đề linh hoạt, trong khi CoAP sử dụng hệ thống khám phá tài nguyên ổn định.
- MQTT không được mã hóa. Thay vào đó, nó sử dụng TLS/SSL (Transport Layer Security/Secure Sockets Layer) để mã hóa bảo mật.
- Rất khó để tạo một mạng MQTT có thể mở rộng toàn cầu.
- Các thách thức khác liên quan đến bảo mật, khả năng tương tác và xác thực.

Do giao thức MQTT không được thiết kế với mục đích bảo mật nên giao thức này thường được sử dụng trong các mạng back-end an toàn cho các mục đích dành riêng cho ứng dụng. Cấu trúc topic của MQTT có thể dễ dàng tạo thành một cây khổng lồ và không có cách rõ ràng nào để chia cây thành các miền logic nhỏ hơn. Điều này gây khó khăn cho việc tạo ra mạng MQTT có thể mở rộng toàn cầu bởi vì kích thước của cây chủ đề tăng lên, độ phức tạp sẽ tăng lên.

Một nhược điểm khác của MQTT là thiếu khả năng tương tác (inter-operability). Bởi vì payload của message là binary, nếu không có thông tin về cách chúng được mã hóa, thì các vấn đề có thể phát sinh – đặc biệt là trong các kiến trúc mở, nơi các ứng dụng khác nhau từ các nhà sản xuất khác nhau hoạt động cùng nhau.

Như đã đề cập trước đây, MQTT có các tính năng xác thực tối thiểu được tích hợp trong giao thức. Tên người dùng và mật khẩu được gửi ở dạng văn bản rõ ràng và bất kỳ hình thức sử dụng MQTT an toàn nào cũng phải sử dụng SSL/TLS, và thật không may, đây không phải là một giao thức nhẹ.

Xác thực ứng dụng khách (client) bằng client-side certificate không phải là một quy trình đơn giản và MQTT không có cách nào để kiểm soát ai sở hữu topic và ai có thể publish thông tin trên đó, ngoại trừ sử dụng các phương tiện độc quyền (proprietary), ngoài băng tần (out-of-band). Điều này giúp dễ dàng đưa các message có hại vào mạng, do cố ý hoặc do nhầm lẫn.

2

Hơn nữa, không có cách nào để người nhận message biết ai đã gửi message gốc trừ khi thông tin đó có trong message. Các tính năng bảo mật phải được triển khai trên MQTT theo kiểu độc quyền sẽ làm tăng code footprint và khiến việc triển khai trở nên khó khăn hơn.

2.2.4.3. ỨNG DỤNG CỦA MQTT

Do các đặc tính nhẹ của nó, MQTT hoạt động tốt cho các ứng dụng liên quan đến giám sát từ xa, bao gồm:

- Đồng bộ hóa các cảm biến, chẳng hạn như đầu báo cháy hoặc cảm biến chuyển động để phát hiện hành vi trộm cắp, để xác định xem mối nguy hiểm có hợp lệ hay không;
- Theo dõi các thông số sức khỏe bằng cảm biến cho các bệnh nhân đã xuất viện;
- Cảm biến cảnh báo con người về sự nguy hiểm.

Một ứng dụng khác là ứng dụng nhắn tin dựa trên văn bản để liên lạc theo thời gian thực, tận dụng mức sử dụng năng lượng và dữ liệu thấp của MQTT. Ví dụ: Facebook sử dụng MQTT cho ứng dụng Messenger của mình, không chỉ vì giao thức này tiết kiệm pin trong khi nhắn tin từ điện thoại đến điện thoại mà còn vì giao thức này cho phép gửi tin nhắn hiệu quả trong một phần nghìn giây, mặc dù kết nối internet không nhất quán trên toàn cầu.

Hầu hết các nhà cung cấp dịch vụ đám mây, bao gồm Amazon Web Services (AWS), Google Cloud, IBM Cloud và Microsoft Azure, đều hỗ trợ MQTT.

MQTT rất phù hợp với các ứng dụng sử dụng thiết bị M2M và IoT cho các mục đích như phân tích thời gian thực, bảo trì và giám sát phòng

ngừa trong các môi trường, bao gồm nhà thông minh, chăm sóc sức khỏe, logistics, công nghiệp và sản xuất.

2

2.2.5. RTSP

Giao thức Real-Time Streaming Protocol (RTSP) được định nghĩa trong RFC 2326 [18]. Đó là một giao thức tầng ứng dụng và chịu trách nhiệm kiểm soát việc giao nhận dữ liệu có tính chất thời gian thực như video hoặc âm thanh trực tiếp.

Giao thức này chịu trách nhiệm về việc thiết lập và kiểm soát phiên làm việc. Nhờ vào giao thức này, máy khách có khả năng phát, tạm dừng và dừng luồng. RTSP hoạt động cùng với các giao thức khác như Real-time Transport Protocol (RTP) và RTP Control Protocol(RTCP), vì RTSP không thể thực hiện việc kiểm soát và vận chuyển luồng. RTSP hoạt động như một bộ điều khiển từ xa mạng cho các máy chủ đa phương tiện.

Các thông điệp RTSP có thể được gửi qua cả TCP hoặc UDP và sử dụng cổng 554 mặc định. Để thực hiện kết nối RTSP, giao tiếp phải hoạt động từ lúc máy khách gửi yêu cầu đến máy chủ, cho đến khi máy khách gửi gói tin cuối cùng của video hoặc âm thanh.

Giao thức RTSP hỗ trợ các hoạt động sau:

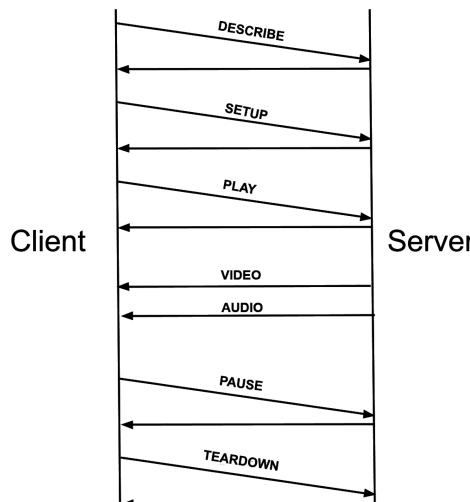
- Truy xuất phương tiện từ máy chủ phương tiện: Máy khách có thể yêu cầu một phiên kết nối qua HTTP hoặc một phương pháp khác.
- Mời máy chủ phương tiện tham gia phiên: Một máy chủ phương tiện có thể được mời tham gia phiên hiện tại, để phát lại hoặc để ghi lại tất cả hoặc một phần của phương tiện trong bài thuyết trình. Chế độ này hữu ích cho các ứng dụng giảng dạy phân tán.
- Thêm phương tiện vào bài thuyết trình hiện tại: Đặc biệt đối với bài thuyết trình trực tiếp, nó hữu ích nếu máy chủ có thể thông báo cho máy khách về phương tiện bổ sung đang trở nên có sẵn.

2.2.5.1. NGUYÊN LÝ HOẠT ĐỘNG CỦA RTSP

- **Thiết lập kết nối(Connection Establishment):** Quy trình làm việc RTSP bắt đầu với việc máy khách thiết lập kết nối với máy chủ phương

tiện, thường sử dụng TCP làm giao thức truyền tải. Máy khách khởi tạo kết nối bằng cách gửi yêu cầu RTSP đến máy chủ, và nhận được phản hồi từ máy chủ.

- **Yêu cầu mô tả phương tiện truyền thông(Media Description Request):** Khi kết nối được thiết lập, máy khách sẽ gửi yêu cầu DESCRIBE để lấy mô tả về tài nguyên phương tiện. Mô tả này thường được cung cấp ở định dạng Session Description Protocol(SDP), bao gồm thông tin về các luồng phương tiện, codec và giao thức truyền tải.



Hình 2.5: Cách thức hoạt động của RTSP [45]

- **Thiết lập luồng(Stream Setup):** Máy khách gửi yêu cầu SETUP để thiết lập các luồng phương tiện và thiết lập các tham số truyền tải cho RTP (Giao thức truyền tải thời gian thực) và RTCP (Giao thức điều khiển RTP). Máy chủ phản hồi với thông tin truyền tải cần thiết, chẳng hạn như địa chỉ IP và số cổng để gửi và nhận các gói RTP và RTCP.
- **Phát lại phương tiện(Media Playback):** Với các luồng phương tiện được thiết lập, khách hàng gửi yêu cầu PLAY để bắt đầu phát lại tài nguyên phương tiện. Máy chủ bắt đầu gửi dữ liệu đa phương tiện bằng RTP và RTCP để truyền dữ liệu phương tiện và kiểm soát thông tin.

- **Điều khiển phương tiện(Media Control):** Trong khi phát lại, máy khách có thể gửi các yêu cầu RTSP bổ sung để kiểm soát luồng phương tiện, chẳng hạn như PAUSE để tạm dừng phát lại hoặc TEARDOWN để chấm dứt phiên và giải phóng tất cả các tài nguyên liên quan.
- **Truyền phương tiện(Media Transmission):** Trong khi phương tiện đang được phát lại, giao thức RTP chịu trách nhiệm truyền dữ liệu phương tiện thực tế (âm thanh và video), trong khi giao thức RTCP được sử dụng để gửi thông tin điều khiển, chẳng hạn như phản hồi về chất lượng của phương tiện nhận được và dữ liệu đồng bộ hóa.
- **Chấm dứt phiên(Session Termination):** Yêu cầu TEARDOWN được gửi đến máy chủ khi quá trình phát lại hoàn tất hoặc máy khách quyết định dừng phiên phát trực tuyến. Máy chủ sau đó giải phóng tất cả các tài nguyên liên quan đến phiên và đóng kết nối.

2.2.5.2. CÁC TRẠNG THÁI - STATES

RTSP điều khiển một luồng dữ liệu có thể được gửi qua một giao thức riêng biệt, độc lập với kênh điều khiển. Ví dụ, việc điều khiển RTSP có thể xảy ra trên một kết nối TCP trong khi dữ liệu truyền qua UDP. Do đó, việc giao nhận dữ liệu vẫn tiếp tục ngay cả khi máy chủ truyền thông không nhận được bất kỳ yêu cầu RTSP nào. Ngoài ra, trong vòng đời của nó, một luồng truyền thông đa phương tiện duy nhất có thể được điều khiển bằng các yêu cầu RTSP được phát ra tuần tự trên các kết nối TCP khác nhau. Do đó, máy chủ cần duy trì "trạng thái phiên" để có thể tương quan các yêu cầu RTSP với một luồng dữ liệu.

Nhiều Method trong RTSP không góp phần vào trạng thái. Tuy nhiên, các Method sau đây đóng vai trò quan trọng trong việc xác định phân bổ và sử dụng tài nguyên luồng trên máy chủ: SETUP, PLAY, RECORD, PAUSE và TEARDOWN.

- **SETUP:** Gây ra việc máy chủ phân bổ tài nguyên cho một luồng và bắt đầu một phiên RTSP.
- **PLAY và RECORD:** Bắt đầu truyền dữ liệu trên một luồng được phân bổ thông qua SETUP.

- PAUSE: Tạm ngừng một luồng mà không giải phóng tài nguyên của máy chủ.

2

- TEARDOWN: Giải phóng tài nguyên liên quan đến luồng. Phiên RTSP ngừng tồn tại trên máy chủ.

2.2.5.3. TRUYỀN PHÁT CAMERA IP BẰNG RTSP

Truyền phát camera IP là truyền video thời gian thực qua Internet Protocol(IP) sử dụng giao thức (RTSP). Nguyên tắc cơ bản đầu sau phát trực tuyến camera IP là sử dụng giao thức RTSP để truyền dữ liệu. Giao thức này cho phép camera IP gửi và nhận dữ liệu thời gian thực với độ trễ và tiêu thụ băng thông thấp.

Việc sử dụng RTSP trong phát trực tuyến camera IP giúp người dùng có thể xem các luồng video trực tiếp từ các vị trí từ xa trên bất kỳ thiết bị nào được kết nối với Internet. Điều này mở ra nhiều khả năng khác nhau, chẳng hạn như giám sát từ xa, hội nghị từ xa,...

Ngoài ra, RTSP cung cấp cho người dùng hình ảnh chất lượng cao hơn so với các giao thức phát trực tuyến khác do các thuật toán nén hiệu quả của nó. Do đó RTSP là một trong những giao thức được camera IP hỗ trợ nhiều nhất cho phát tuyến.

2.2.6. HTTP

2.2.6.1. TỔNG QUAN GIAO THỨC HTTP

HTTP (HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản) là một trong những giao thức chuẩn trên Internet, sử dụng để truyền tải thông tin giữa Máy chủ web và Máy khách web. Đây là giao thức Client/Server được sử dụng trên World Wide Web (WWW).

HTTP là một ứng dụng của bộ giao thức TCP/IP, cung cấp khả năng trao đổi dữ liệu giữa các thiết bị trên mạng.

2.2.6.2. NGUYÊN LÝ HOẠT ĐỘNG CỦA HTTP

HTTP hoạt động dựa trên mô hình Client – Server, với máy tính người dùng làm máy khách (Client). Khi người dùng thực hiện một hành động, máy khách gửi yêu cầu đến máy chủ (Server) và đợi câu trả lời.

HTTP là một giao thức stateless, có nghĩa là mỗi yêu cầu không biết về các yêu cầu trước đó. Điều này cho phép truyền tải dữ liệu độc lập và xây dựng hệ thống không phụ thuộc vào dữ liệu trước đó.

2.2.6.3. CÁC THÀNH PHẦN CHÍNH CỦA HTTP

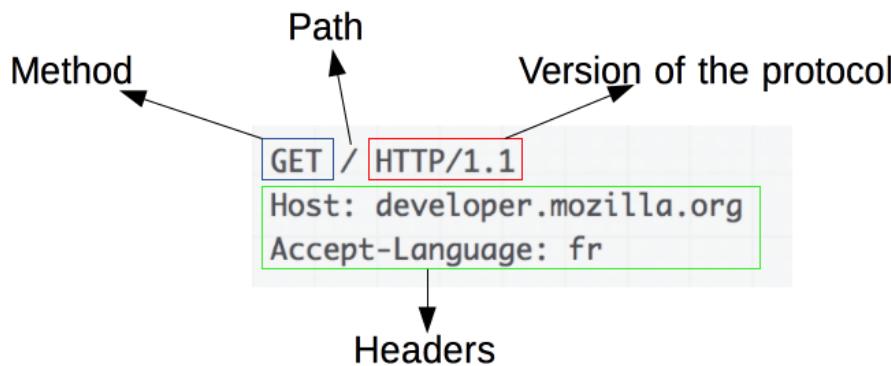
HTTP - Requests

HTTP Request Method: Là phương thức để chỉ ra hành động mong muốn được thực hiện trên tài nguyên đã xác định.

- Một Request-line = Phương thức + URI–Request + Phiên bản HTTP. Giao thức HTTP định nghĩa một tập các giao thức **GET, POST, HEAD, PUT, DELETE, OPTIONS...** Client có thể sử dụng một trong các phương thức đó để gửi request lên server.
- Có thể có hoặc không các trường header
- Một dòng trống để đánh dấu sự kết thúc của các trường Header.
- Request Header Fields: Các trường header cho phép client truyền thông tin bổ sung về yêu cầu, và về chính client, đến server. Một số trường: Accept-Charset, Accept-Encoding, Accept-Language, Authorization, Expect, From, Host,

Khi request đến server, server thực hiện một trong 3 hành động sau:

- Phân tích request và trả lại tài nguyên yêu cầu cho client.
- Phân tích request, thực thi chương trình và trả lại kết quả.
- Trả lại thông báo lỗi nếu không thể đáp ứng yêu cầu.



Hình 2.6: HTTP request [6]

HTTP - Responses

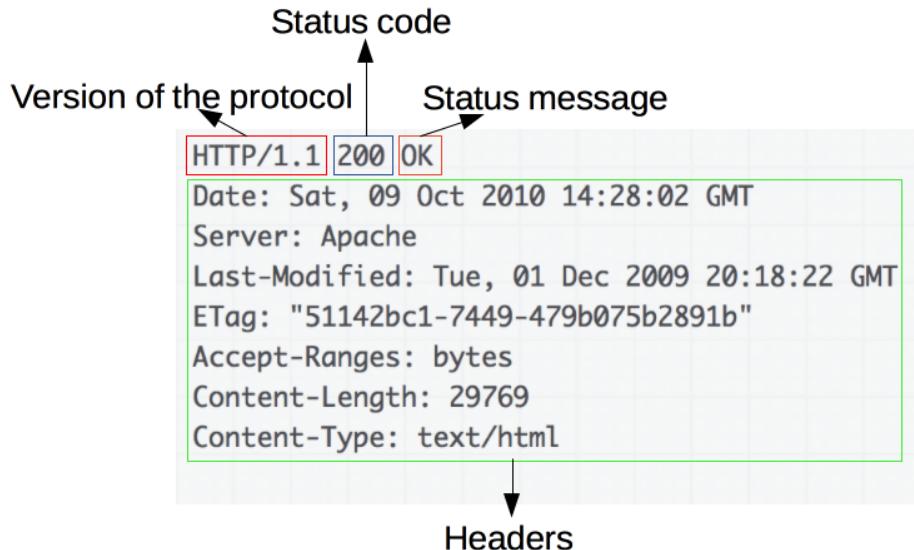
Cấu trúc của một HTTP response:

- Một Status Line bao gồm phiên bản giao thức (HTTP-Version) sau đó là mã hóa trạng thái số (Status-Code) và cụm từ thuần văn bản được liên kết của nó.
- Có thể có hoặc không có các trường header
- Một dòng trống để đánh dấu sự kết thúc của các trường header
- Mã trạng thái: Thông báo về kết quả khi nhận được yêu cầu và xử lý bên server cho client.

2.2.6.4. MÃ TRẠNG THÁI TRONG HTTP

- 1xx (100 – 199): Thông tin - Yêu cầu đã được chấp nhận và đang được xử lý.
- 2xx (200 – 299): Thành công - Yêu cầu đã được máy chủ tiếp nhận và xử lý thành công.
 - 200 OK: Yêu cầu thành công.

2



Hình 2.7: HTTP response [6]

- 201 Created: Yêu cầu thành công và tạo mới một tài nguyên.
 - 202 Accepted: Yêu cầu được nhận nhưng chưa được thực hiện hoàn toàn.
 - 204 No Content: Không có nội dung để gửi, header có thể cập nhật cache.
 - 205 Reset Content: Yêu cầu này yêu cầu user-agent đặt lại document đã gửi.
 - 206 Partial Content: Dùng khi client yêu cầu chỉ một phần của nguồn tài nguyên.
3. 3xx (300 – 399): Điều hướng - Client cần thực hiện hành động bổ sung để hoàn thành yêu cầu.
- 301 Moved Permanently: URL của tài nguyên đã thay đổi vĩnh viễn.
 - 303 See Other: Điều hướng client lấy nguồn từ URI khác với yêu cầu GET.
 - 304 Not Modified: Dùng cho caching, client sử dụng phiên bản cache hiện tại.

4. 4xx (400 – 499): Lỗi phía client - Yêu cầu không hợp lệ hoặc chứa cú pháp sai.

- 400 Bad Request: Máy chủ không hiểu yêu cầu do cú pháp không hợp lệ.
- 401 Unauthorized: Client chưa được xác thực.
- 402 Payment Required: Code hiếm khi sử dụng, thường để hệ thống thanh toán kỹ thuật số.
- 403 Forbidden: Client không có quyền truy cập.
- 404 Not Found: Server không tìm thấy tài nguyên.
- 405 Method Not Allowed: Phương thức yêu cầu bị vô hiệu hóa.

5. 5xx (500 – 599): Lỗi phía máy chủ - Máy chủ không thể hoàn thành yêu cầu hợp lệ.

- 500 Internal Server Error: Lỗi chung khi máy chủ gặp phải trường hợp bất ngờ.
- 501 Not Implemented: Máy chủ không nhận biết hoặc không thể xử lý phương thức yêu cầu.
- 503 Service Unavailable: Máy chủ hiện không sẵn sàng hoặc quá tải, trạng thái tạm thời.

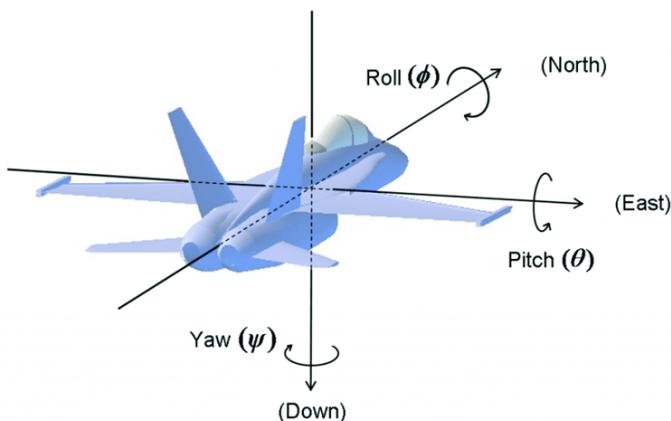
2.3. TOẠ ĐỘ VÀ GÓC TRONG KHÔNG GIAN

2.3.1. TOẠ ĐỘ EULER

2

Góc Euler là ba góc được giới thiệu bởi Leonhard Euler để miêu tả định hướng của một vật thể rắn. Để mô tả như một hướng trong không gian ba chiều Euclidean, cần ba tham số. Chúng tôi giới thiệu và sử dụng toạ độ Euler như một bước đầu tiên để mô tả và quản lý các khía cạnh xoay của cánh tay robot cũng như các đối tượng liên quan trong Unity 3D.

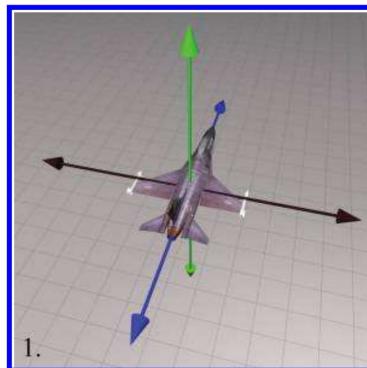
Trong toạ độ Euler, người ta thường định nghĩa các hướng bằng góc yaw, pitch và roll. Một số tài liệu khác có thể định nghĩa bằng heading, pitch, bank tương ứng. Với bộ số (roll, pitch, yaw) được cho, ta có thể



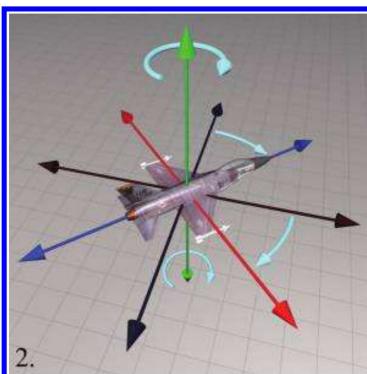
Hình 2.8: Hình ảnh về trục roll, pitch, yaw trong không gian 3D [14]

biểu diễn một vật thể trong không gian 3D theo 4 bước như sau:

- **Bước 1:** Bắt đầu ở hướng "nhận dạng" - nghĩa là, với các trục không gian đối tượng được căn chỉnh với các trục đứng.
- **Bước 2:** Hiện thực góc xoay yaw, xoay quanh trục y, được thể hiện như hình. Giá trị dương thì xoay sang bên phải (theo chiều kim đồng hồ khi nhìn từ trên xuống)
- **Bước 3:** Góc pitch mô tả cho việc xoay quanh trục x. Trục x ta đang nói là trục x trong không gian đối tượng, không phải trục x trong



Hình 2.9: Vị trí ban đầu của vật thể theo góc Euler [14]



Hình 2.10: Xoay vật thể theo trục y [14]

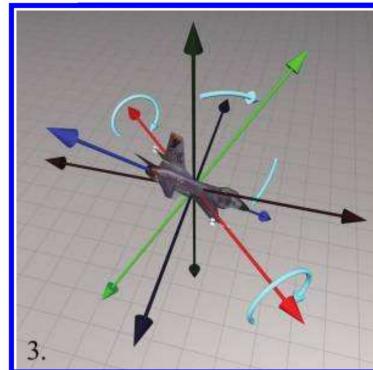
không gian thế giới thật (hay ta thường tưởng tượng nó nằm ngang). Nói cách khác, pitch sẽ ước lượng cho ta góc nghiêng theo hướng Bắc-Nam của một vật thể trong không gian.

- **Bước 4:** Góc roll mô tả cho việc xoay quanh trục z. Trục z ta đang nói cũng là trục z trong không gian đối tượng. Nói cách khác, roll sẽ ước lượng cho ta góc nghiêng theo hướng Đông-Tây của một vật thể trong không gian.

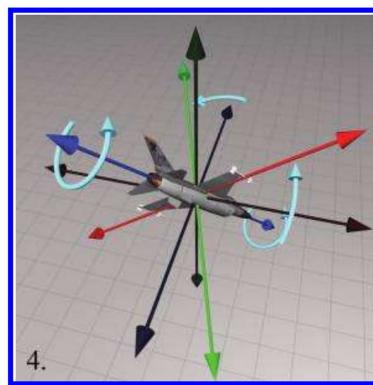
2.3.1.1. ƯU ĐIỂM

- Dễ sử dụng và tiếp cận: bởi vì bộ ba góc Euler là các góc, giống như suy nghĩ của con người từ trước đến giờ về góc và hướng. Ví dụ, hãy

2



Hình 2.11: Xoay vật thể theo trục x trong không gian đối tượng [14]



Hình 2.12: Xoay vật thể theo trục z trong không gian đối tượng [14]

tưởng tượng bạn đang nhìn thấy một cái bàn. Nếu bạn nghĩ về việc xoay nó ngang, điều này có thể được miêu tả bằng một góc 90° . Sự dễ sử dụng này là một ưu điểm quan trọng. Khi một hướng cần được hiển thị dưới dạng số hoặc nhập vào từ bàn phím, thì góc Euler thực sự là lựa chọn duy nhất.

- Góc Euler biểu diễn hướng của đối tượng bằng cách sử dụng ít giá trị nhất có thể: góc Euler chỉ cần 3 số để mô tả một hướng trong không gian. Không có phương pháp nào khác có thể sử dụng ít hơn ba số để biểu diễn định hướng trong không gian 3D. Điều này trở nên quan trọng khi bộ nhớ của hệ thống có giới hạn, vì vậy Góc Euler trở thành lựa chọn tối ưu nhất để biểu diễn hướng của đối tượng.

- Bất kỳ bộ ba số nào cũng là hợp lệ: nếu chúng ta chọn ngẫu nhiên 3 số, chúng luôn tạo thành một bộ ba góc Euler hợp lệ mà chúng ta có thể hiểu như một biểu diễn về định hướng. Nói cách khác, không có khái niệm về bộ ba góc Euler không hợp lệ. Tất nhiên, các số có thể không biểu diễn hướng một cách chính xác, nhưng ít nhất chúng là hợp lệ.

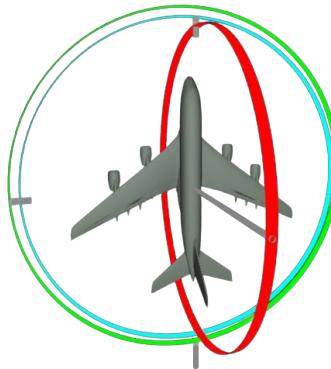
2.3.1.2. NHƯỢC ĐIỂM

Nhược điểm chính của góc Euler bao gồm ba vấn đề như sau:

- Sự không nhất quán trong cách biểu diễn hướng: ta có thể nhận ra rằng, với một định hướng cho trước, ta có thể biểu diễn nó bằng nhiều bộ ba Euler khác nhau, điều này tạo ra sự không rõ ràng và khó xác định khi muốn đưa ra một biểu diễn duy nhất, tương tự như hiện tượng aliasing trong xử lý tín hiệu. Ví dụ, khi có một bộ ba góc Euler (Roll, Pitch, Yaw) như $(30^\circ, 60^\circ, 45^\circ)$, không chỉ có một cách để biểu diễn nó. Một biểu diễn khác có thể là $(390^\circ, 420^\circ, 405^\circ)$, trong đó các góc được thay đổi nhưng vẫn mô tả cùng một hướng trong không gian. Sự đa dạng này trong biểu diễn góc làm tăng độ phức tạp và gây khó khăn trong việc đảm bảo tính duy nhất và thống nhất của mô tả hướng.
- Vấn đề nội suy (interpolating) giữa hai định hướng: nghĩa là 3 góc Euler không hoàn toàn độc lập với nhau. Ví dụ, khi xoay pitch 1 góc 135° , điều này tương đương với xoay yaw 180° , sau đó xoay pitch xuống góc 45° , và cuối cùng là xoay roll 180° . Mặc dù các góc đã thay đổi, nhưng kết quả cuối cùng vẫn là một định hướng trong không gian.
- Vấn đề gimbal lock: Vấn đề gimbal lock xảy ra khi góc xoay của pitch bằng $\pm 90^\circ$, việc đó khiến cho trục xoay của pitch và yaw trùng với nhau, từ đó làm cho đối tượng trong không gian mất đi 1 bậc tự do và làm giảm khả năng biểu diễn một số hướng quay trong không gian.

Hình ảnh dưới đây mô tả tình trạng gimbal lock xảy ra, với trục pitch là màu xanh biển (cyan) trùng với trục yaw (xanh lá), khi đó khiến cho việc xoay trục yaw và roll là như nhau.

2



Hình 2.13: Hình ảnh minh họa về gimbal lock

2.3.2. TOẠ ĐỘ QUATERNION

Vấn đề của chính của Euler như ta đã đề cập ở trên là về gimbal lock, toạ độ quaternion sẽ giải quyết vấn đề này bằng cách sử dụng 4 số để biểu diễn hướng của một vật thể trong không (vì vậy nó được đặt tên là quaternion).

2.3.2.1. KÍ HIỆU TRONG QUATERNION

Trong quaternion sẽ đại lượng vô hướng chỉ độ lớn và đại lượng còn lại là vector 3D, ta gọi thành phần vô hướng là w và phần vector 3D là v hoặc (x, y, z)

Quaternions có thể được biểu diễn dưới hai dạng sau:

$$[w \ v], \quad [w \ (x \ y \ z)]$$

2.3.2.2. ÁP DỤNG QUATERNION VÀO PHÉP QUAY KHÔNG GIAN

Giả sử ta có một vector trục quay u (vector u phải là vector đơn vị, tức $u_x^2 + u_y^2 + u_z^2 = 1$) và góc quay θ (dương khi quay cùng chiều với vector, theo nguyên tắc nắm tay phải), chúng ta sẽ có một quaternion quay như sau [28]:

$$\mathbf{q} = \cos \frac{1}{2}\theta + \mathbf{u} \sin \frac{1}{2}\theta = \cos \frac{1}{2}\theta + (u_x \mathbf{i} + u_y \mathbf{j} + u_z \mathbf{k}) \sin \frac{1}{2}\theta$$

2

Ví dụ, ta muốn tạo một phép quay một góc 120 độ (tức $\frac{2}{3}\pi$ radian) quay trục $\mathbf{v} = \mathbf{i} + \mathbf{j} + \mathbf{k}$. Đầu tiên, chúng ta sẽ chuẩn hóa (normalize) vector \mathbf{v} thành vector đơn vị \mathbf{u} .

$$\mathbf{u} = \hat{\mathbf{v}} = \frac{\mathbf{v}}{|\mathbf{v}|} = \frac{\mathbf{i} + \mathbf{j} + \mathbf{k}}{\sqrt{1^2 + 1^2 + 1^2}} = \frac{1}{\sqrt{3}}(\mathbf{i} + \mathbf{j} + \mathbf{k})$$

Và từ đó, chúng ta có quaternion cho phép quay là:

$$\mathbf{q} = \cos \frac{1}{2}\theta + \mathbf{u} \sin \frac{1}{2}\theta = \cos \frac{\pi}{3} + \frac{1}{\sqrt{3}}(\mathbf{i} + \mathbf{j} + \mathbf{k}) \sin \frac{\pi}{3} = \frac{1 + \mathbf{i} + \mathbf{j} + \mathbf{k}}{2}$$

Để thực hiện phép quay, ta cần một cái gì đó để quay. Ví dụ, ta có một điểm P trong không gian có tọa độ (a, b, c) được xác định trong không gian bởi vector $\mathbf{p} = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$ thì phép quay điểm P theo \mathbf{q} được định nghĩa là:

$$\mathbf{p}' = \mathbf{q} \mathbf{p} \mathbf{q}^{-1}$$

Trong đó \mathbf{p}' là vector tọa độ của điểm P sau khi xoay. Và \mathbf{q}^{-1} là nghịch đảo của \mathbf{q} . Cụ thể trong trường hợp này:

$$\mathbf{q}^{-1} = \frac{1}{\mathbf{q}} = \cos \frac{1}{2}\theta - \mathbf{u} \sin \frac{1}{2}\theta = \frac{1 - \mathbf{i} - \mathbf{j} - \mathbf{k}}{2}$$

Sau đó, chúng ta sẽ sử dụng phép nhân quaternion. Hãy ghi nhớ phép nhân quaternion không có tính giao hoán. Cho 2 quaternion $\mathbf{q}_1 = w_1 + x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k}$ và $\mathbf{q}_2 = w_2 + x_2 \mathbf{i} + y_2 \mathbf{j} + z_2 \mathbf{k}$, thì tích $\mathbf{q} = \mathbf{q}_1 \mathbf{q}_2$ là:

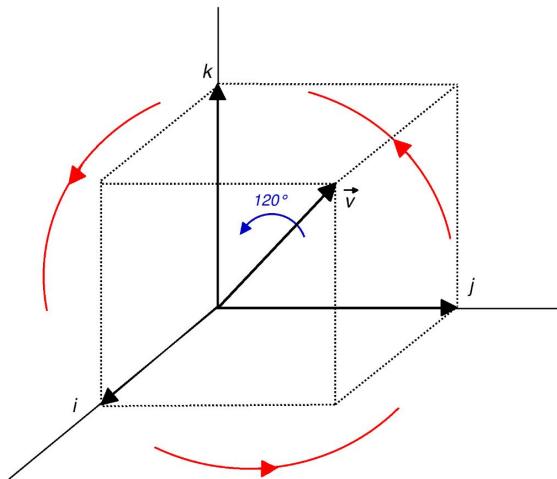
2

$$\begin{aligned}\mathbf{q} = & w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2 \\ & + (w_1 x_2 + x_1 w_2 + y_1 z_2 - z_1 y_2) \mathbf{i} \\ & + (w_1 y_2 - x_1 z_2 + y_1 w_2 + z_1 x_2) \mathbf{j} \\ & + (w_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 w_2) \mathbf{k}\end{aligned}$$

Sử dụng công thức trên để áp vào ví dụ của chúng ta (biến đổi $\mathbf{p} = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$ theo \mathbf{q}), ta được

$$\mathbf{p}' = \mathbf{q} \mathbf{p} \mathbf{q}^{-1} = ci + aj + bk$$

Hình dưới sẽ mô tả phép quay quaternion. Chúng ta có thể thấy nếu chúng ta quay quanh trục v một góc 120 độ thì tọa độ (a, b, c) sẽ tráo cho nhau thành (c, a, b) .



Hình 2.14: Ví dụ về phép quay quaternion trong không gian [28]

2.3.3. TOA ĐỘ HƯỚNG VÀ GÓC NGHỈENG CỦA CAMERA

Phần này sẽ mô tả cụ thể hướng nhìn của camera trong không gian được xác định như thế nào cụ thể bằng 2 giá trị đo lường Azimuth và Elevation.

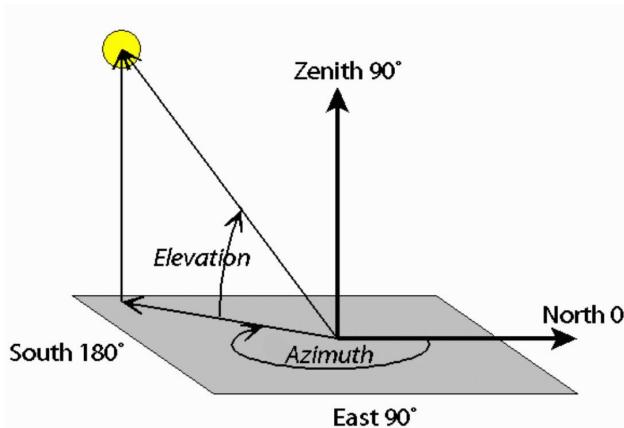
2.3.3.1. HƯỚNG (AZIMUTH) CỦA CAMERA

Azimuth là giá trị đo lường giúp ta xác định góc giữa hướng Bắc so với vật thể. Giá trị của Azimuth thay đổi từ 0° đến 360° . Nó bắt đầu từ phía Bắc ở 0° . Khi ta quay sang phải (theo chiều kim đồng hồ) tức giá trị Azimuth lúc đó 90° , sau đó là phía Nam (180°), tiếp theo là phía Tây (270°), và sau đó quay trở lại phía Bắc (360° cũng như 0°).

2.3.3.2. ĐỘ CAO (ELEVATION) CỦA CAMERA

Elevation thường chỉ độ cao của camera so với mặt đất. Điều này quan trọng để xác định góc nhìn và vùng phủ sóng của camera. Giá trị của Elevation thay đổi từ 0° đến 90° .

Với ảnh minh họa dưới đây, vòng tròn màu vàng đại diện cho vật thể. Nó có góc azimuth là khoảng 200° và góc elevation cỡ 60° (bằng $2/3$ so với góc tối đa của nó là 90° so với bầu trời).



Hình 2.15: Ví dụ về góc azimuth và elevation [46]

3

THIẾT KẾ VÀ HIỆN THỰC HỆ THỐNG

Thiết kế và hiện thực hệ thống là một phần không thể thiếu trong quá trình phát triển các hệ thống điều khiển và tự động hóa. Việc thiết kế và hiện thực hệ thống đảm bảo rằng hệ thống sẽ hoạt động đúng như các yêu cầu đặt ra. Do đó, việc thiết kế và hiện thực hệ thống là rất quan trọng trong quá trình phát triển và có vai trò quyết định đến sự thành công của dự án. Trong phần tiếp theo sau đây sẽ trình bày về quy trình thiết kế và hiện thực hệ thống điều khiển một cánh tay robot thông qua giao thức MQTT và điều khiển camera dựa trên giao thức HTTP cũng như cách ta có thể lấy được một stream video từ IPCamera dựa trên câu lệnh trong giao thức RTSP.

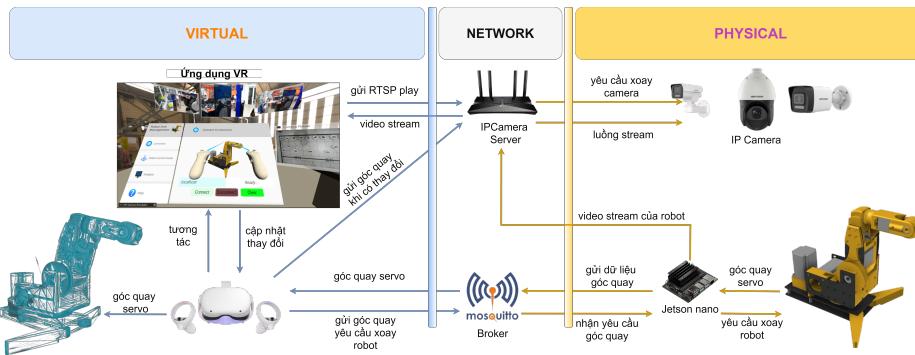
3.1. TỔNG QUAN HỆ THỐNG

Ở phần tổng quan, nhóm sẽ đưa ra kiến trúc hệ thống mà nhóm sử dụng cho việc hiện thực ứng dụng điều khiển, đồng thời liệt kê một số thiết bị, công nghệ được ứng dụng trong đồ án này.

3.1.1. KIẾN TRÚC HỆ THỐNG

3

Trong dự án này, nhóm sẽ thiết kế mô hình kiến trúc hệ thống gồm 3 lớp như sau:



Hình 3.1: Kiến trúc ba lớp gồm **virtual** (lớp ảo), **network** (mạng), và **physical** (vật lý). Lớp virtual tương tác với người dùng và cung cấp trải nghiệm thực tế ảo. Lớp network quản lý giao tiếp giữa các thành phần ảo và vật lý trong hệ thống. Cuối cùng, lớp physical bao gồm các thiết bị và tài nguyên vật lý như mô hình robot, jetson nano và ipcamera.

- **Lớp ảo (virtual):** Lớp này có nhiệm vụ tạo ra một môi trường thực tế ảo (Virtual Reality - VR) cho người dùng. Trong môi trường VR người dùng có thể điều khiển mô hình 3D của cánh tay robot bằng một tay cầm (sẽ giới thiệu ở phần sau), các tín hiệu điều khiển của các khớp (joint) sẽ được publish lên feed "servo_control" thông qua giao thức MQTT và để hiển thị thông số góc quay các khớp của Robot, ta cần subscribe vào topic "servo_data" để nhận thông tin về góc quay. Ngoài ra trong môi trường VR cũng có 3 màn hình video được hiển thị, các khung hình này lấy từ 3 IPCamera được tích hợp xung quanh thông qua giao thức RTSP. Điều này giúp người dùng có khả năng quan sát môi trường thực tế để có một cái nhìn rõ ràng hơn về nơi làm việc.

- Lớp mạng (network): Lớp này có nhiệm vụ quản lý giao tiếp giữa các thành phần ảo và vật lý trong hệ thống. Router đóng vai trò trung tâm xử lý thông tin và chuyển tiếp dữ liệu giữa các thành phần khác nhau. Trong trường hợp Oculus Quest 2, router đóng vai trò trung gian trong việc chuyển tiếp các yêu cầu HTTP giữa thiết bị và IP Camera. Oculus Quest 2 gửi yêu cầu HTTP chứa thông số azimuth và elevation khi người dùng xoay đầu. Router nhận được yêu cầu này, sau đó chuyển tiếp yêu cầu đến IP Camera. IP Camera sẽ thực hiện xoay theo các thông số azimuth và elevation được cung cấp để xoay về góc tương ứng. Ngoài ra, router còn xử lý các lệnh RTSP cho việc lấy khung cảnh từ IPCamera và camera của Robot về Unity.
- Lớp vật lý (physical): Lớp vật lý (physical): Lớp này bao gồm cánh tay robot, IPCamera và Jetson Nano (một máy tính nhúng AI, sẽ được giải thích chi tiết ở phần sau). Cánh tay robot đóng vai trò chính, thực hiện các chuyển động dựa trên các tương tác được điều khiển từ lớp ảo. Jetson Nano sẽ subscribe vào feed "servo_control" để nhận thông tin về các tín hiệu điều khiển của từng khớp và thực hiện yêu cầu xoay cho cánh tay thật. Ngoài ra, để hiển thị thông số góc quay của servo trong thế giới ảo như đã nói trên, thông tin sẽ được publish vào feed "servo_data". IPCamera chịu trách nhiệm cung cấp video stream cho lớp ảo thông qua giao thức RTSP, tái tạo môi trường xung quanh cánh tay trong môi trường VR.

3.1.2. GIỚI THIỆU THIẾT BỊ VÀ CÔNG NGHỆ

3.1.2.1. JETSON NANO VÀ 6DoF ROBOT ARM

Jetson Nano là một bo mạch phát triển dành cho trí tuệ nhân tạo (AI) và học sâu (Deep Learning), được phát triển bởi NVIDIA. Với kích thước nhỏ gọn, Jetson Nano được trang bị một bộ vi xử lý ARM Cortex-A57 64 bit và một đồng xử lý đồ họa NVIDIA Maxwell GPU với 128 nhân CUDA. Jetson Nano cũng có khả năng xử lý hình ảnh và video với tốc độ cao, cho phép nó được sử dụng để phát triển các ứng dụng nhận dạng hình ảnh, phân loại, giám sát và điều khiển trong các lĩnh vực như xe tự lái, robot, y tế và an ninh. Bo mạch phát triển này cũng được trang bị các cổng kết nối như Ethernet, USB, HDMI và GPIO, cho phép người dùng kết nối với các thiết bị ngoại vi khác như camera, màn hình, bàn phím và chuột. NVIDIA cũng cung cấp một loạt các công cụ và phần mềm hỗ trợ để phát

triển các ứng dụng trên Jetson Nano, bao gồm JetPack SDK, DeepStream SDK và NVIDIA TensorRT. Nó cũng được hỗ trợ bởi các thư viện phổ biến như TensorFlow, PyTorch và Caffe. Jetson Nano là một lựa chọn tuyệt vời cho các nhà phát triển, sinh viên và những người đam mê công nghệ AI và học sâu. Nó cung cấp một cách dễ dàng để khám phá và phát triển các ứng dụng trí tuệ nhân tạo đa dạng.



3

Hình 3.2: Hình ảnh Jetson Nano

Trong đồ án này, Jeson Nano được sử dụng như một gateway với giao thức MQTT. Jeson Nano chịu trách nhiệm nhận và đẩy dữ liệu qua lớp áo, hỗ trợ điều khiển cánh tay Robot Arm 6DoF thông qua việc truyền thông bằng giao thức MQTT, đảm bảo việc truyền thông tin hiệu quả và đáng tin cậy giữa các thành phần của hệ thống.

Robot Arm 6DoF là một robot đang được phát triển bởi một dự án DIY, robot này gồm có 6 khớp. Robot này cung cấp khả năng nhận dạng hình ảnh, xử lý dữ liệu và điều khiển chính xác, cho phép người dùng thực hiện nhiều tác vụ khác nhau.

Robot Arm 6DoF có thiết kế linh hoạt, với khung thép không gỉ và

bốn bộ phận cơ khí được điều khiển bởi các động cơ servo. Tay cầm này cung cấp nhiều khả năng chuyển động, cho phép người dùng di chuyển và xoay cánh tay cầm để đạt được các vị trí khác nhau. Robot Arm 6DoF cũng được trang bị camera có độ phân giải cao và khả năng xử lý hình ảnh, cho phép người dùng thực hiện các tác vụ nhận dạng đối tượng, phân loại, giám sát và điều khiển.

3

Robot Arm 6DoF là một lựa chọn tuyệt vời cho các nhà phát triển, sinh viên và những người đam mê công nghệ AI và robot. Nó cung cấp một cách dễ dàng để khám phá và phát triển các ứng dụng trí tuệ nhân tạo và robot với khả năng điều khiển chính xác và độ tin cậy cao. Dưới đây là hình ảnh về cánh tay Robot:



Hình 3.3: Hình ảnh cánh tay Robot Arm 6DoF ngoài đời thật

Dưới đây là về dữ kiện về góc quay và thông tin của từng khớp cánh tay robot:

- **Khớp 1:** Góc quay từ -90° đến 90° , là phần nối từ trực xoay bên dưới và phần thân của cánh tay.



Hình 3.4: Hình ảnh cánh tay Robot Arm 6DoF ngoài đời thật

- **Khớp 2:** Góc quay từ -70° đến 70° , là phần khớp kết nối đầu tiên ở khung cố định trên phần cánh tay.
- **Khớp 3:** Góc quay từ -60° đến 180° , là phần khớp giữ cánh tay ở trên cùng
- **Khớp 4:** Góc quay từ -90° đến 90° , là phần kê cạnh khớp của cánh tay, nơi phụ trách chuyển động của đầu cánh tay
- **Khớp 5:** Góc quay từ -90° đến 90° , là phần đầu của cánh tay, có dạng hình vuông
- **Khớp 6:** Góc quay tùy ý, cuối cùng là khớp được gắn trên phần đầu ở khớp 5, phụ trách việc tương tác với vật thể

3.1.2.2. OCULUS QUEST 2

Oculus Quest 2 là một thiết bị thực tế ảo (VR) độc lập, có thể được sử dụng mà không cần kết nối với một máy tính hoặc điện thoại thông minh. Nó được phát triển bởi công ty Oculus, một công ty con của Facebook, và ra mắt vào năm 2020.

3

Oculus Quest 2 được trang bị hai màn hình OLED độ phân giải 1832 x 1920 pixel trên mỗi mắt, cung cấp hình ảnh sắc nét và màu sắc chân thực. Nó cũng được trang bị bộ xử lý Snapdragon XR2, cung cấp hiệu suất mạnh mẽ để chạy các trò chơi và ứng dụng VR.

Với khả năng theo dõi chuyển động 6 độ tự do, Oculus Quest 2 cho phép người dùng di chuyển tự do trong không gian thực tế ảo, tạo ra trải nghiệm tuyệt vời cho người dùng. Nó cũng đi kèm với hai bộ điều khiển Oculus Touch để kiểm soát và tương tác với thế giới ảo.

Trong đồ án này, Oculus Quest 2 được sử dụng trong vài trò là 1 client cho người dùng tương tác với mô hình 3D của cánh tay được xây dựng trong Unity, người dùng có thể điều khiển và sử dụng các tính năng AI được phát triển trên cánh tay robot thông qua Oculus.



Hình 3.5: Hình ảnh Oculus Quest 2

3.1.2.3. CAMERA

3.1.2.4. IPCAMERA

Giám sát các hoạt động trong nhà máy cần tích hợp nhiều camera ở các khu vực khác nhau giúp cho người quản lý dễ dàng quan sát từ xa mà không cần đến trực tiếp khu vực đó. Nhóm đang sử dụng ba camera IP đến từ hãng Hikvision. Ở hình bên dưới, camera ở giữa là cam 360 độ cho ta góc nhìn tùy chỉnh để giám sát được ở nhiều góc độ trong nhà máy.



Hình 3.6: Hình ảnh 3 camera

Ba camera có các thông số chính có nội dung như sau:

Camera	Bên Trái	Giữa	Bên Phải
Main Stream 60Hz	24fps(2560×1440)	30fps(1920 × 1080)	
Compression	Main stream: H.265+/H.265/H.264+/H.264		
VideoBitRate (Kbps to Mbps)	32 to 8	32 to 16	32 to 8
Protocols	TCP/IP, HTTP, RTP, RTSP, NTP, IGMP, IPv6, UDP, QoS, FTP, SMTP, ICMP, DHCP, DNS		

Cả ba camera trên là camera IP nên ta có thể lấy luồng phát trực tiếp theo cú pháp sau:

rtsp://<username>:<password>@<IP address of device>:<RTSP port>/Streaming/channels/<channelnumber><stream number>

Những tham số này khi cấu hình camera sẽ có được, chỉ cần điền tham số tương ứng là ta có thể có đường dẫn như sau, ví dụ: **rtsp://admin :bkcamera2023@192.168.8.50:554/Streaming/channels/101**

Dưới đây là hình ảnh thực tế, ba camera gắn trực tiếp trên một trụ với ba hướng khác nhau giúp tăng tối đa không gian quan sát. Kết nối tới Modem WIFI để tạo thành mạng cục bộ. Ta chỉ cần kết nối chung vào mạng cục bộ này, sử dụng đường dẫn ở trên là lấy được nội dung trực tiếp.

3



Hình 3.7: Hình ảnh ba camera được lắp đặt thực tế

3.1.2.5. ROBOT WEBCAM

Để có thể theo dõi cụ thể hơn các góc nhìn toàn cảnh, trên robot cũng có tích hợp một webcam.

Để ứng dụng thực tế ảo có thể lấy stream từ webcam robot, ta cũng cần stream video đó lên RTSP server và từ đó, ứng dụng có thể lấy stream về.

Công nghệ sử dụng để stream RTSP trên nền tảng robot (linux) là sử dụng GStreamer:

GStreamer [17] là một framework cho phép chúng ta có thể tạo ra

các ứng dụng đa phương tiện (multimedia). Một trong số các ứng dụng multimedia có thể kể đến là: Trình phát video và nhạc (media player), Streaming server, ứng dụng chỉnh sửa video (Video editor) hoặc có thể kể đến một ứng dụng hỗ trợ chat video đang được sử dụng rất nhiều trong thời gian Covid-19 (ví dụ như Google meet, Microsoft Team...) thì GStreamer cũng có thể dùng để tạo ra được những ứng dụng tương tự như vậy.

3.1.2.6. OPENCV

OpenCV là một thư viện mã nguồn mở cho xử lý ảnh và computer vision, được phát triển bởi Intel vào năm 1999. Thư viện này cung cấp các hàm xử lý ảnh và video, bao gồm các thuật toán như xử lý hình ảnh, lọc, phân tích, nhận dạng đối tượng, theo dõi chuyển động và phát hiện biến độ.

OpenCV hỗ trợ các ngôn ngữ lập trình như C++, Python và Java, và có sẵn trên các nền tảng khác nhau bao gồm Windows, Linux, macOS và các thiết bị di động như iOS và Android.

Các tính năng chính của OpenCV bao gồm:

- Xử lý ảnh và video: OpenCV cung cấp các thuật toán xử lý ảnh và video như là lọc, biến đổi, cân bằng màu sắc, sửa chữa ảnh hỏng và phát hiện chuyển động.
- Phân tích và nhận dạng đối tượng: OpenCV cung cấp các thuật toán để phân tích và nhận dạng đối tượng trong ảnh, bao gồm các thuật toán như phát hiện khuôn mặt, phát hiện đối tượng và nhận dạng ký tự.
- Theo dõi chuyển động: OpenCV cho phép theo dõi chuyển động của đối tượng trong video, bao gồm các thuật toán như theo dõi Kalman và theo dõi quan sát.

OpenCV được sử dụng rộng rãi trong các ứng dụng thực tế, bao gồm robot, xe tự hành, xử lý ảnh y tế, xử lý video và công nghệ nhận diện khuôn mặt. Ở module Control, nhóm sử dụng OpenCV để nhận diện màu sắc vật thể trong hai chức năng AI là Waste_classification và Object_tracking

3

Hình 3.8: OpenCV

3.1.2.7. UNITY 3D

Unity 3D là một công cụ phát triển game và ứng dụng tương tác 3D đa nền tảng. Được phát triển bởi Unity Technologies, Unity 3D có khả năng phát triển trò chơi cho nhiều nền tảng như iOS, Android, Windows, Mac, PlayStation, Xbox và Nintendo Switch.

Unity 3D cho phép các nhà phát triển tạo ra các trò chơi và ứng dụng tương tác 3D đẹp mắt và chân thực, với khả năng tùy chỉnh cao, đa dạng các tính năng và sử dụng các công nghệ hiện đại như AI, VR và AR. Điều này đặc biệt hữu ích cho các ứng dụng thương mại, giáo dục, tuyên truyền, kỹ năng, tâm lý học, v.v.

Unity 3D cũng hỗ trợ một loạt các công cụ và tài nguyên, bao gồm các công cụ thiết kế và phát triển game, thư viện đồ họa, âm thanh, quản lý nội dung, và nhiều hơn nữa. Điều này giúp các nhà phát triển và nhà thiết kế dễ dàng tạo ra các trò chơi và ứng dụng tương tác 3D chất lượng cao.

Với cộng đồng phát triển đông đảo, Unity 3D là một công cụ được ưa chuộng và được sử dụng rộng rãi trên toàn thế giới.

Trong đề tài này, nhóm sử dụng Unity3D để tạo ra môi trường thực tế ảo tăng cường tương tác với mô hình 3D của cánh tay



3

Hình 3.9: Unity3D

3.2. ỨNG DỤNG THỰC TẾ ẢO (VR APPLICATION)

Digital Twin - Bản sao ảo là một khái niệm đang dần nổi lên trong thời đại công nghiệp 4.0, và càng phát triển mạnh mẽ hơn dưới sự tăng tiến của công nghệ IoT, IIoT, AI, Big Data... Digital Twin có nhiều ứng dụng trong mô phỏng, quản lý dân cư, quản lý nhà thông minh, lĩnh vực sức khỏe và đặc biệt là công nghiệp. Digital Twin được xem là bước tiến mới trong công nghiệp, đem lại nhiều lợi ích cho nhà sản xuất, nhất là khả năng kiểm soát sản xuất, dự báo rủi ro và bảo trì sớm để hạn chế thời gian ngắt (down-time) trong sản xuất, tối ưu nguồn lợi nhuận. Trong quá trình nghiên cứu, nhóm nhận ra hai yếu tố không thể thiếu trong các ứng dụng digital twin, đó là cách thức truyền nhận dữ liệu hai chiều thời gian thực và một mô hình ảo tương ứng với vật thể thực tế. Ở phần này, nhóm sẽ tập trung nhiều vào cách các nhà nghiên cứu khác hiện thực một Digital Twin của họ. Trong nghiên cứu hiện thực một digital twin cho hệ thống thang máy [31], Marouane Ouadoudi sử dụng Matlab/simulink chạy ứng dụng Digital Twin. Cụ thể, mô hình thang máy được tạo từ SolidWorks dưới dạng CAD model, sau đó chuyển đổi sang dạng file phù hợp để đưa vào Matlab/Simulink. Nhờ vào việc này, mô hình digital twin không chỉ thu thập dữ liệu nhờ giao tiếp OPC UA, còn có khả năng mô phỏng trong môi trường Matlab. Mika Lohtader [25] xây dựng một mô hình 3D cho mục tiêu phát triển Digital Twin của một đơn vị sản xuất nhỏ (Micro Manufacturing Unit). Tác giả tạo mô hình sử dụng SolidWorks kèm theo các thông tin vật lý và chức năng cần thiết. Mô hình này sau đó được chuyển đổi thành FlexSim Simulation model nhằm thực hiện những mô phỏng thời gian thực cho Digital Twin. FlexSim còn tích hợp khả năng xây dựng ứng dụng thực tế ảo (Virtual Reality - VR) và cho

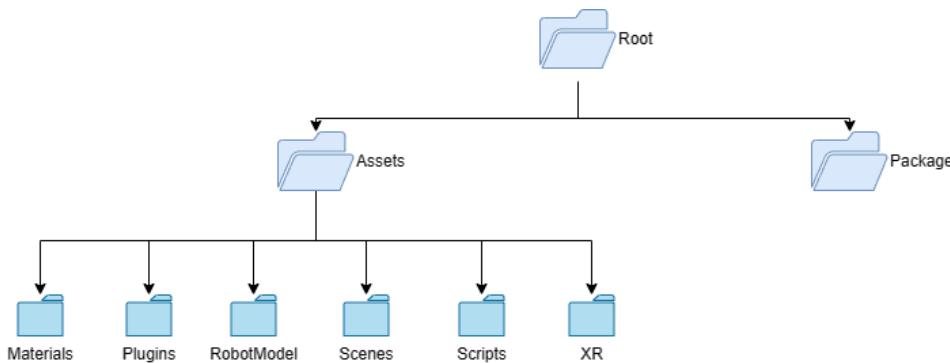
phép người dùng tương tác với các thành phần của mô hình. Amirashkan Haghshenas [19] lựa chọn Unity 3D cho việc tạo dựng ứng dụng digital Twin của hệ thống cối xoay gió ngoài khơi. Theo tác giả, Unity Engine là một nền tảng phát triển ứng dụng game dễ sử dụng, có thể xây dựng trên nhiều nền tảng, thân thiện và dễ lập trình với ngôn ngữ C#. Sử dụng Unity, tác giả đã tạo nên hai ứng dụng: một ứng dụng 3D cho mô phỏng với 4 nguồn dữ liệu khác nhau: Dữ liệu tương tác người dùng, dữ liệu thực tế từ OPC UA, dữ liệu mô phỏng từ model Matlab và dữ liệu lịch sử được lấy từ internet. Ứng dụng thứ hai là sự mở rộng cho ứng dụng thứ nhất, cho phép người dùng sử dụng thiết bị di động để kích hoạt thực tế ảo tăng cường (Augmented Reality - AR) và tương tác với digital twin thay vì trên máy tính. Các ứng dụng digital twin khá đa dạng và có thể được phát triển sử dụng nhiều loại phần mềm khác nhau, nhưng điểm chung là các phần mềm này luôn cố gắng tạo ra một bảo sao ảo cá nhân hóa của vật thể / thực thể, cung cấp các thông tin, dữ liệu trạng thái của vật thể và thêm nữa là hỗ trợ mô phỏng, nghiên cứu thêm dựa trên dữ liệu thời gian thực được thu thập.

Trong đồ án này, nhóm cũng quyết định sử dụng Unity Engine làm công cụ chính để phát triển ứng dụng trong môi trường thực tế ảo VR trên kính Oculus Quest 2 và điều khiển thiết bị thông qua joystick, cùng với 3 camera như đã nói, tạo thành một không gian 3D giúp đem lại trải nghiệm điều khiển và quan sát thú vị, chân thực hơn cho người dùng. Với sự hỗ trợ của thư viện trên ngôn ngữ C#, nhóm dễ dàng hiện thực một MQTT client trên ứng dụng nhằm kết nối tới MQTT broker và thu thập dữ liệu cũng như điều khiển. Hiện tại, ứng dụng chỉ thể hiện trạng thái thực tế của cánh tay với độ trễ thấp và độ trễ RTT với mức độ vừa phải (1s) đồng thời cho phép người dùng thực hiện điều khiển cánh tay robot. Nhóm chưa có phương pháp phù hợp để phát triển ứng dụng có khả năng mô phỏng, và dự định sẽ phát triển thêm khả năng này trong tương lai.

3.2.1. CÁCH TỔ CHỨC CHƯƠNG TRÌNH VR APPLICATION

Như đã trình bày ở phần trước, VR app được thiết kế sử dụng game engine Unity phiên bản **2022.3.5f1** với ngôn ngữ C#. Cấu trúc thư mục của ứng dụng được thể hiện như hình bên dưới:

Các thư mục chứa các tài nguyên của Unity 3D như sau:



3

Hình 3.10: Cấu trúc thư mục của VR Application trong Unity 3D

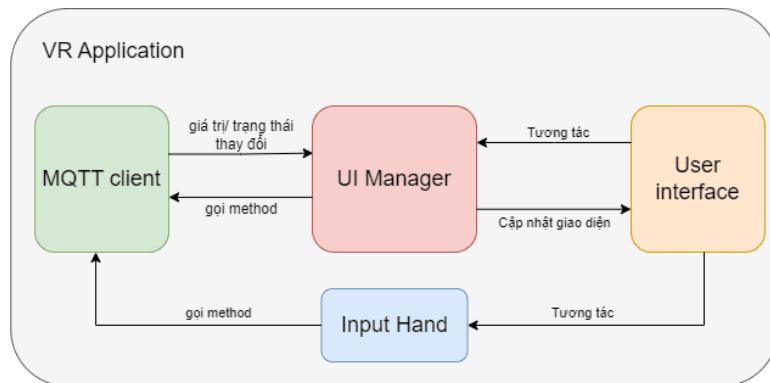
- **Assets:** nơi lưu trữ các tài nguyên của ứng dụng. Bao gồm các tệp, hình ảnh, âm thanh, material, shader, script, scene. Ta có thể sử dụng công cụ Asset Store để tìm kiếm và tải xuống các tài nguyên từ cộng đồng Unity hoặc từ các nhà phát triển khác để tích hợp vào trong ứng dụng.
- **Packages:** Chứa 2 tệp là manifest.json and package-lock.json để quản lý thông tin về phiên bản cụ thể của gói. Tệp manifest.json là tệp luôn ở gốc của gói và có thông tin quan trọng về gói, như tên và phiên bản của gói. Tệp package.json là tệp mô tả chi tiết nội dung của gói. Khi một dự án Unity 3D được mở Unity Package Manager sẽ đọc tệp này và tải các gói này xuống. Ta có thể xem nội dung các gói này trực tiếp trên giao diện Unity 3D.

3.2.2. HIỆN THỰC MODULE VR APPLICATION

Để thiết kế ứng dụng chạy trong môi trường VR, nhóm sử dụng Asset Oculus Integration được tải về từ Unity Asset Store và tích hợp vào quá trình phát triển. Sơ đồ khái quát của module được thể hiện như dưới:

Cấu trúc trong ứng dụng VR có thể chia làm ba phần chính:

- MQTT client: để thực hiện kết nối và pub/sub đến MQTT Broker.
- Khối tiếp nhận và xử lý tương tác (gồm UI Manager, InputHand) sẽ tiếp nhận các tương tác đưa xuống từ giao diện và thực hiện việc



3

Hình 3.11: Sơ đồ khái niệm của VR app

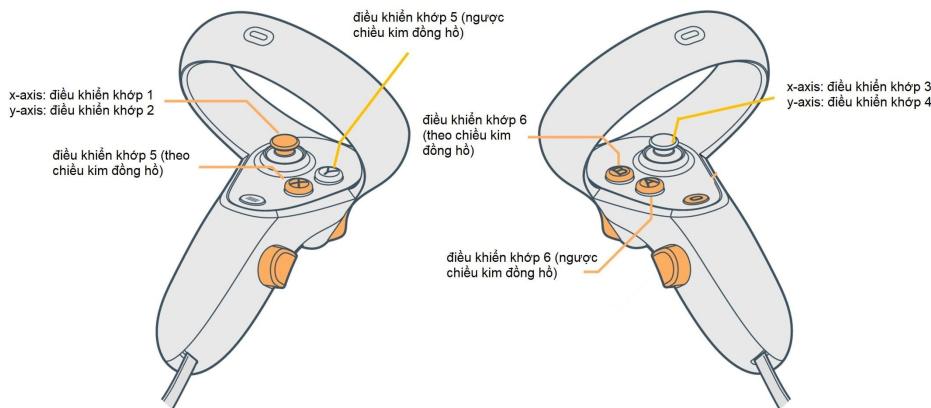
gọi method phù hợp để publish data đến MQTT handler.

- Khôi giao diện ứng dụng, bao gồm nút nhấn, mô hình robot 3D, và màn hình chứa ba camera theo dõi thực tế. Phần này tạo ra không gian VR trong ứng dụng, cho phép người dùng tương tác và theo dõi diễn biến thực tế như trong không gian ảo.

Phần tiếp theo trình bày chi tiết hơn chức năng của từng khôi:

- MQTT client**: Khôi này cung cấp các hàm để kết nối/ ngắt kết nối tới MQTT server, subscribe/publish vào các feed như "servo_data", "servo_control", "robot_mode".
- UI_manager**: Quản lý một phần giao diện điều khiển cơ bản của ứng dụng. Khôi này tồn tại như một thành phần trung gian giữa các đối tượng Unity (Object) và MQTT handler. Trên giao diện ứng dụng, khi người dùng tương tác với các nút nhấn, UI_manager cung cấp các hàm trung gian được "gắn kết" với các nút nhấn này, qua đó gọi method để publish dữ liệu xuống MQTT client và thực hiện điều khiển. Ở chiều ngược lại, khôi quản lý UI này cũng sẽ liên tục lấy dữ liệu về trạng thái cánh tay robot được lưu ở feed "servo_data", và cập nhật lên giao diện những thông tin như:
 - Góc xoay hiện tại của cánh tay robot
 - Tốc độ điều khiển hiện tại

- **InputHand:** Đây là khối "nhận" và "xử lý" các sự kiện nhấn nút của người dùng trên hai tay cầm điều khiển (controller) của Oculus Quest 2. Khi "nhận" và "xử lý" được các sự kiện này, ta có thể tùy vào sự kiện để làm một số hành động nào đó. Trong hiện thực của nhóm, các sự kiện nhấn nút trên tay cầm điều khiển sẽ được dùng để điều khiển cánh tay robot xoay qua, lại, tối, lui, lên, xuống. Khác với 2 khối trên, khối này chỉ có 1 chiều điều khiển tới khối MQTT handler. Chức năng của từng nút nhấn như sau:



Hình 3.12: Chú thích chức năng điều khiển sử dụng tay cầm Oculus Quest 2

- Tay cầm bên trái:
 - ◊ **Joystick x-axis:** Điều khiển khớp 1 của cánh tay máy Robot.
 - ◊ **Joystick y-axis:** Điều khiển khớp 2 của cánh tay máy Robot.
 - ◊ **Nút X:** Xoay khớp thứ 5 cánh tay theo chiều kim đồng hồ.
 - ◊ **Nút Y:** Xoay khớp thứ 5 cánh tay ngược chiều kim đồng hồ.
- Tay cầm bên phải:
 - ◊ **Joystick x-axis:** Điều khiển khớp 3 của cánh tay máy Robot.
 - ◊ **Joystick y-axis:** Điều khiển khớp 4 của cánh tay máy Robot.
 - ◊ **Nút B:** Xoay khớp thứ 6 cánh tay theo chiều kim đồng hồ.
 - ◊ **Nút A:** Xoay khớp thứ 6 cánh tay ngược chiều kim đồng hồ.
- **Giao diện ứng dụng:** Sau cùng là khối giao diện, là nơi trang trí, sắp xếp các mô hình, không gian, nút nhấn, nhân vật... tạo nên giao diện của ứng dụng. Unity cho phép ta thiết kế giao diện nhanh

chóng bằng cách kéo thả các đối tượng Unity (Object hay game Object) trong không gian 3 chiều của 1 bối cảnh (Scene), qua đó giảm bớt sự khó khăn trong thiết kế giao diện ứng dụng. Các đối tượng Unity có thể được tùy chỉnh, thêm chức năng, hình ảnh và đặc biệt là "gắn" các script C# để thực hiện chuỗi các chức năng mà ta hiện thực.

3

Phần tiếp theo sẽ trình bày một số chức năng chính của ứng dụng VR. Nhóm sẽ diễn giải các chức năng này chủ yếu thông qua sequence diagram. Đối với chức năng phức tạp hơn sẽ có kèm theo flowchart để giải nghĩa.

3.2.3. HIỆN THỰC CHỨC NĂNG LIÊN QUAN ĐẾN CAMERA

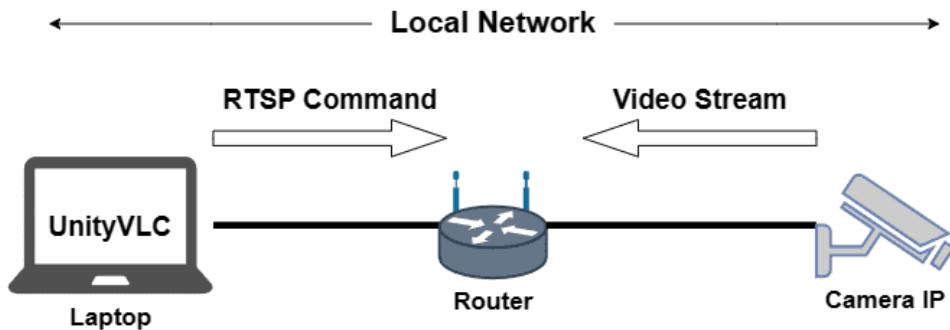
3.2.3.1. LẤY DỮ LIỆU TỪ CAMERA IP VÀ HIỂN THỊ LÊN VR APPLICATION

Trong ứng dụng của nhóm bao gồm ba màn hình để quan sát không gian xung quanh của nhà máy. Ba màn hình chiếu nội dung tương ứng với ba camera thực tế. Hikvision Camera đã cung cấp URL mặc định để có thể lấy được nội dung trực tiếp. [22]

Ba camera kết nối chung tới mạng cục bộ và được cấu hình ba địa chỉ IP khác nhau tương ứng. Ta có thể dùng phần mềm SADP của Hikivision để cấu hình địa chỉ IP và thiết lập các cài đặt cơ bản cho camera [35]. Các thiết bị muốn lấy nội dung trực tiếp cần phải kết nối chung vào cùng mạng cục bộ. Ta có thể dùng ứng dụng VLC trên Laptop để kiểm tra trước URL đã thực sự hoạt động hay chưa.

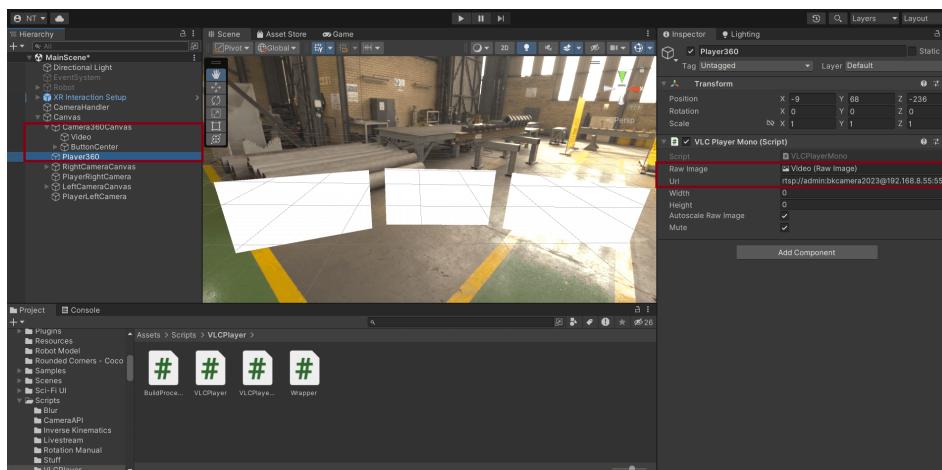
Ở đây nhóm tham khảo thư viện "Libvlc player for unity" [24] hỗ trợ lấy nội dung trực tiếp qua giao thức RTSP. Các bước để tích hợp thư viện có thể tham khảo đường dẫn GitHub trên. Tiếp theo ta cần tạo GameObject Canvas và GameObject RawImage là nơi hiển thị nội dung được lấy từ camera. Tạo thêm GameObject và gắn VLCPlayerMono.cs scripts.

Ta kéo RawImage vừa tạo vào ô RawImage bên cửa sổ Inspectors, và thay đổi đường dẫn trong ô URL thành đường dẫn lấy nội dung trực tiếp của Camera. Như vậy khi Play nội dung từ camera sẽ được hiển thị trong ứng dụng.



3

Hình 3.13: Sơ đồ kết nối mạng cục bộ với camera

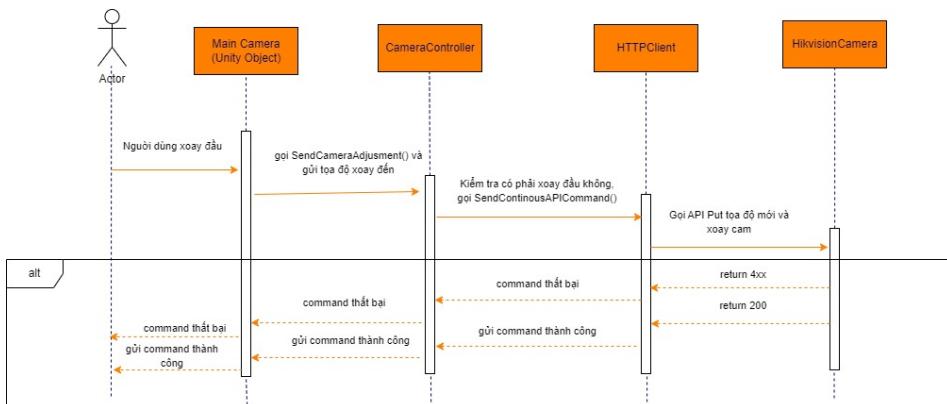


Hình 3.14: Tạo GameObject và cấu hình để hiển thị nội dung trực tiếp trong Unity 3D

3.2.3.2. CHỨC NĂNG XOAY ĐẦU TRONG MÔI TRƯỜNG VR ĐỂ ĐIỀU KHIỂN CAMERA HIKVISION

Chức năng xoay đầu trong môi trường thực tế ảo (VR) là một tính năng quan trọng giúp người dùng tương tác trực tiếp với camera Hikvision và cảm nhận thế giới thực một cách sống động, chân thật nhất.

Nếu người dùng xoay đầu khi sử dụng kính VR, quá trình xử lý sẽ bao gồm việc lấy góc quay của camera. Sau đó, chuyển đổi từ góc Euler sang Azimuth và Elevation để định rõ toạ độ cần quay của camera. Cuối cùng, thông qua việc gửi request bằng giao thức HTTP và sử dụng method `SendCameraAdjustmentCommand()`, ta sẽ gửi một XMLHttpRequest đến



Hình 3.15: Sequence diagram gửi request xoay đầu của kính VR của camera Hikvision

URI: <http://ipAddress/ISAPI/PTZCtrl/channels/<ID>/continuous>, với XML body chứa hai thông số elevation và azimuth được đề cập ở phần 4.8 với format như sau [21].

```

<?xml version="1.0" encoding="UTF-8"?>
<PTZData>
<pan> <!-- opt, xs:integer, -100..100 --> </pan>
<tilt> <!-- opt, xs:integer, -100..100 --> </tilt>
<zoom> <!-- opt, xs:integer, -100.. 100--> </zoom>
</PTZData>

```

Giải thích thuật toán xoay đầu camera

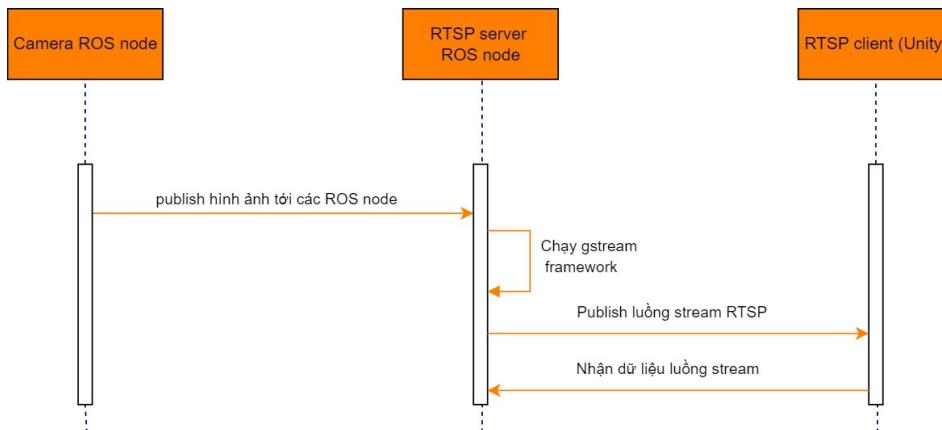
- Đầu tiên khi bắt đầu chạy Unity, ở hàm Start() ta sẽ xác thực người dùng có quyền gọi API đến Camera, và khởi chạy 2 luồng định kì
 - Luồng 1: Kiểm tra độ xoay của kính khi đang đeo trên đầu người dùng có chênh lệnh quá một khoảng 0.01f đến 90f hay không, đây là giá trị được lấy dưới dạng trị tuyệt đối của sự chênh lệch của rotation angles tại vị trí hiện tại so với vị trí cũ. Nếu chỉ là đầu rung không chuyển động thì vào trạng thái IDLE, không xoay camera.
 - Luồng 2: Khởi chạy SendCameraAdjusment và lặp lại mỗi 0.01f. Hàm này thực hiện kiểm tra vị trí mà người dùng đang hướng

tới, xác định vị trí, trạng thái hiện tại, nếu chênh lệch không quá lớn thì trạng thái sẽ vẫn giữ nguyên. Khi trạng thái hiện tại đã đổi ta thực hiện gọi `SendContinuousAPICommand()`.

- Ở `SendContinuousAPICommand()`, đây là một hàm gửi tốc độ xoay và vị trí xoay đầu người dùng hướng đến, thực hiện gọi HTTP Request qua method Put và camera sẽ bắt đầu quay khi API trả về kết quả là thành công.

3

3.2.3.3. HIỆN THỰC WEBCAM ROBOT STREAM RTSP



Hình 3.16: Sequence diagram stream RTSP từ webcam robot tới môi trường Unity

Định nghĩa: ROS Node [34] là một quá trình thực hiện tác vụ cụ thể. Các Node có thể giao tiếp Message (dữ liệu) với nhau bằng cách sử dụng publish/subscribe (truyền/nhận liên tục) hoặc Service (máy khách/máy chủ).

Việc stream webcam của robot tới Unity cụ thể như sau: webcam sẽ có một ROS node mở nó lên, liên tục publish hình ảnh tới các node khác, RTSP server node ở đây có nhiệm vụ lắng nghe tới node của camera, và stream các hình liên tục lên RTSP server, từ đó ta có thể thiết lập được luồng stream, và Unity có thể lấy luồng stream đó thông qua rtsp url được định nghĩa.

3.2.4. HIỆN THỰC CHỨC NĂNG LIÊN QUAN LÊN QUAN GIAO TIẾP MQTT GIỮA ỨNG DỤNG VR VÀ CÁNH TAY ROBOT

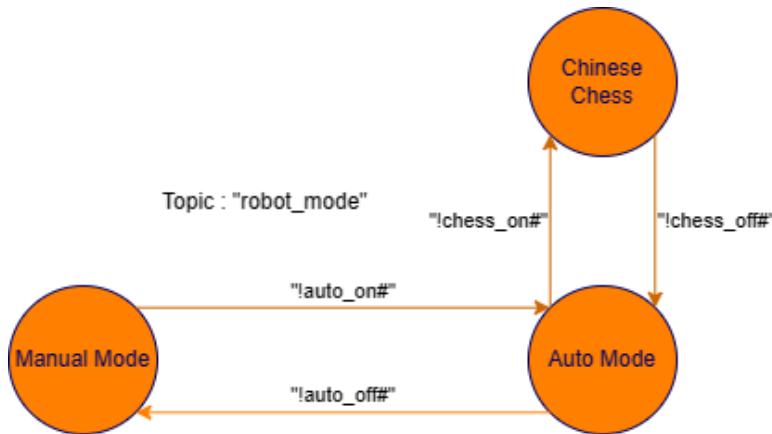
3.2.4.1. ỨNG DỤNG VR KẾT NỐI TỚI CÁNH TAY ROBOT THÔNG QUA GIAO THỨC MQTT

3

Ứng dụng VR là một MQTT Client và Jetson là một MQTT Mosquitto Broker. Ở đây phần hiện thực MQTT Client nhóm tích hợp thư viện "M2MQTT for Unity" [26]. Nhóm sẽ quy định các topic và message cho việc giao tiếp giữa hai bên.

Cánh tay robot có ba trạng thái chính để tương tác với ứng dụng VR:

- Trạng thái điều khiển chuyển động cánh tay Robot thủ công, nhóm gọi là Manual Mode.
- Trạng thái tự động thực hiện chức năng cụ thể, nhóm gọi là Auto Mode.
- Trạng thái thực hiện đánh cờ trong Auto Mode.



Hình 3.17: Trang thái tương tác ứng dụng VR và cánh tay Robot

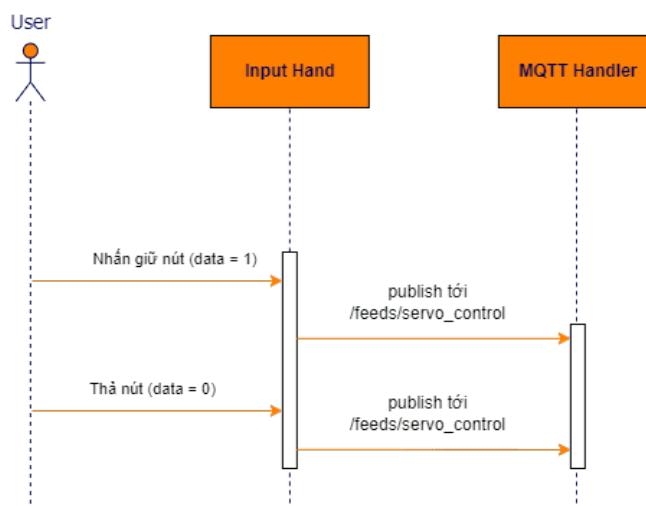
Chuyển đổi trạng thái từ Manual Mode qua Auto Mode, ứng dụng VR gửi message "!auto_on#" tới topic "robot_mode". Chiều ngược lại nhóm gửi message "!auto_off#".

Khi đang ở trong Auto Mode để thực hiện chức năng đánh cờ tướng,

Ứng dụng VR gửi message "!chess_on#" tới topic topic "robot_mode". Ngược lại khi tắt chức năng này gửi message "!chess_off#".

3.2.4.2. CHỨC NĂNG ĐIỀU KHIỂN CÁNH TAY THEO SERVO TRONG MANUAL MODE

Chức năng tiếp theo cho phép người dùng điều khiển cánh tay chuyển động sử dụng 2 tay cầm của Oculus Quest 2. Chi tiết nút nhấn và chức năng của chúng đã được trình bày ở phần trên. Ở đây, sequence diagram của chức năng mô tả các thành phần tham gia vào chức năng và tương tác của chúng:



Hình 3.18: Sequence diagram của chức năng điều khiển cánh tay theo servo

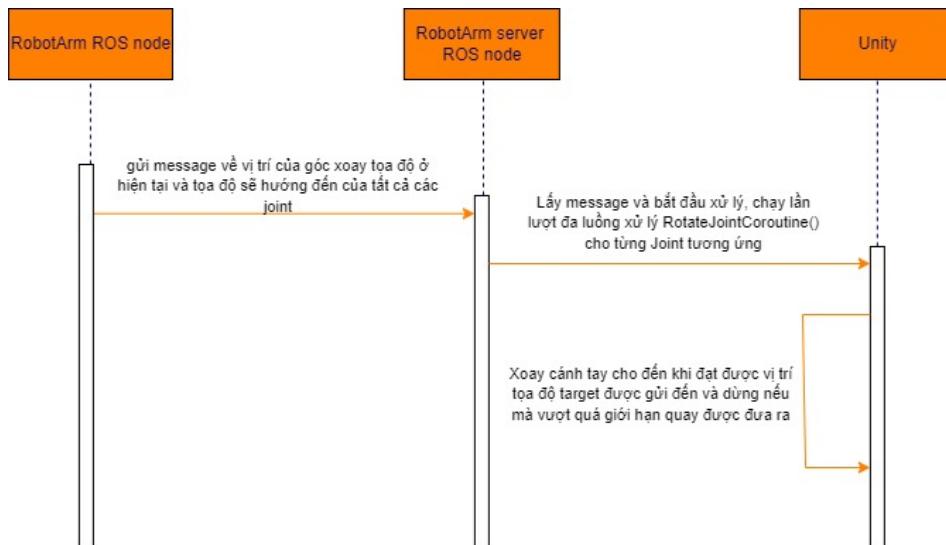
Ở chế độ điều khiển bằng servo, khi người dùng nhấn nút điều khiển, khôi InputHand sẽ "nhận" và "xử lý" sự kiện nhấn, thả nút và publish data đến feed "servo_control", từ feed này, cánh tay thật sẽ subscribe vào và xoay các khớp với dữ liệu nhận được. Đối với nhấn giữ nút, servo sẽ liên tục xoay cho tới khi người dùng thả nút ra. Điều này tạo ra một liên kết động liên tục giữa người dùng và cánh tay robot, cho phép điều khiển mượt mà và linh hoạt theo ý muốn.

Người dùng có thể chuyển đổi qua lại giữa Manual Mode và Auto Mode trong bảng điều khiển. Trong Auto mode, ứng dụng được tích hợp chức năng chơi cờ tướng, khi bật chức năng này lên thì cánh tay robot sẽ

tự động thực hiện chức năng đã được lập trình sẵn. Do đó cần phải đồng bộ chuyển động giữa cánh tay thật và cánh tay ảo.

3.2.4.3. HIỆN THỰC XỬ LÝ ĐỒNG BỘ HÓA CHUYỂN ĐỘNG Ở TRẠNG THÁI AUTOMODE CHO CÁNH TAY ROBOT TRÊN UNITY

Dưới đây là Sequence Diagram về chức năng tự động khi cánh tay robot thật thay đổi vị trí tọa độ các khớp, tín hiệu sẽ được gửi đến lên cánh tay robot trong Unity và thực hiện xoay tương ứng.



Hình 3.19: Sequence diagram stream RTSP từ webcam robot tới môi trường Unity

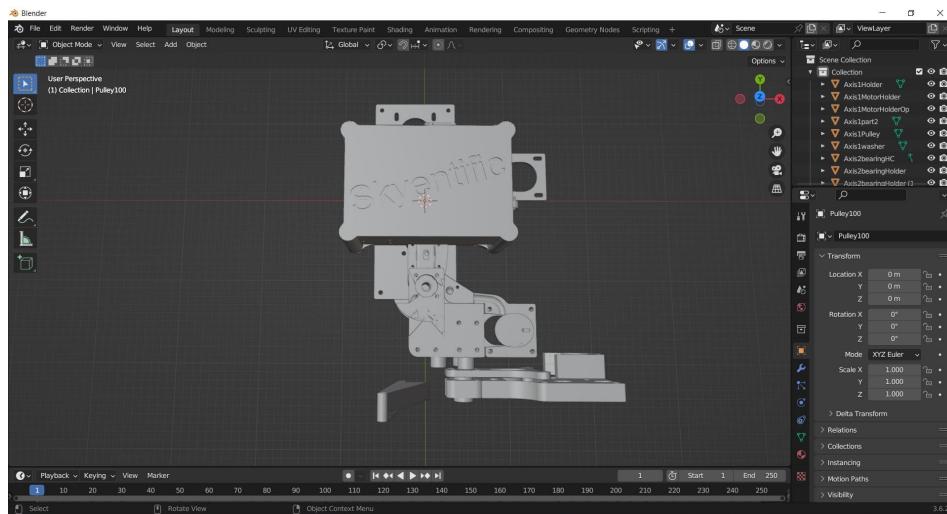
3.2.5. HIỆN THỰC VỀ GIAO DIỆN ỨNG DỤNG

3.2.5.1. IMPORT MÔ HÌNH 3D CỦA CÁNH TAY VÀO UNITY

Mô hình Robot Arm 6DoF 3D là một open source, ta có thể tải các thành phần trong model 3D của robot ở đây [38].

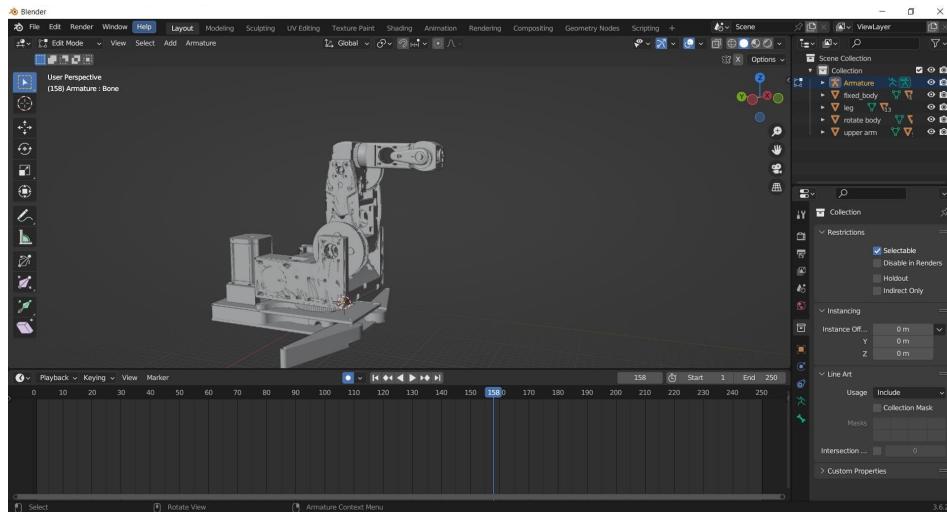
Hiện tại Unity 3D không hỗ trợ trực tiếp import file .stl do đó cần sử dụng ứng dụng trung gian là Blender để chuyển đổi sang file .fbx.

Có thể thấy mô hình cánh tay robot trong blender không được hoàn



Hình 3.20: Mô hình 3D của cánh tay trong Blender

chỉnh, các khớp còn lộn xộn và việc của ta cần làm là xây dựng lại model hoàn chỉnh từ các thành phần riêng biệt, chi tiết các khớp được nối với nhau như nào thì ta có thể tham khảo tại file **SmallRobotArmMechanicalAssembly.pdf** trong tài liệu này [38].



Hình 3.21: Mô hình 3D hoàn chỉnh của cánh tay trong Blender

Sau khi đã xây dựng được mô hình hoàn chỉnh, phần tiếp theo ta sẽ

tìm cách để import model này vào Unity. Ta cần xuất ra file .fbx từ Blender bằng cách File > Export > .fbx. Sau đó ta chỉ cần vào Unity và Import lại file tương ứng.

3

Sau khi import hoàn tất, ta sẽ thực hiện thêm các thao tác để dọn dẹp model, thêm material cho màu sắc model 3D giống với robot thật nhất và nối các khớp và tùy chỉnh góc quay của các khớp trong Unity. Sau một khoảng thời gian dài thử sai và gặp rắc rối với các khớp xoay do chính Unity cũng chỉ cung cấp các component hỗ trợ khớp xoay chỉ có một mối quan hệ cha - con (parent - child), nhóm đã đảm bảo được mô hình 3D trong Unity có thể xoay khớp ổn định ở một mức độ chấp nhận được và mượt mà.



Hình 3.22: Mô hình 3D của cánh tay trong Unity 3D

3.2.5.2. PACKAGE VÀ ASSETS ĐƯỢC SỬ DỤNG CHÍNH

Package chính được sử dụng

Ngoài những Package mặc định, project Unity của nhóm sử dụng những package trong Unity registry sau đây để hỗ trợ trong quá trình phát triển:



Hình 3.23: Mô hình 3D của cánh tay trong Unity 3D

- VR: Oculus XR Plugin [23] và OpenXR Plugin [30]
- Engineering: Visual Studio Editor, JetBrains Rider Editor, Editor Coroutines, Profile Analyzer, Test Framework, Code Coverage.
- 2D Sprite
- Animation Rigging
- Burst
- Input System
- Mathematics
- Unity UI [42]
- XR Hands [20]
- XR Interaction Toolkit [47]
- XR Plugin Management

Trong đó, Oculus XR Plugin và XR Plugin Management là 2 package dùng để tùy chỉnh Unity 3D cho mục đích xây dựng ứng dụng VR. Chi tiết tùy chỉnh các cài đặt trong Unity 3D để xây dựng ứng dụng cho Oculus có thể tham khảo tại đây [12]

Nhóm có sử dụng thêm OpenXR Plugin, XR Hands và XR Interaction Toolkit hỗ trợ nhóm mô phỏng ứng dụng trực tiếp trên Unity 3D mà không cần phần cứng thực tế.

3

Assets được sử dụng trong Project

Ngoài package được tích hợp trong quá trình phát triển, nhóm sử dụng thêm các asset khác được tải từ Unity Asset Store để hỗ trợ tạo hiệu ứng cho giao diện ứng dụng như: DOTween tạo hiệu ứng chuyển động [13], mượt mà hơn, UI Blur [41] tạo hiệu ứng làm mờ, Progress Bar [33] tạo hiệu ứng hiển thị góc quay của khớp ...

4

ĐÁNH GIÁ KẾT QUẢ

Trong chương này, chúng tôi sẽ trình bày chi tiết về các kết quả đạt được trong quá trình triển khai thực hiện đề tài, từ đó đưa ra những kết luận về những thành tựu đạt được, cũng như những thách thức và hạn chế gặp phải trong quá trình thực hiện. Các kết quả đạt được của đề tài sẽ cung cấp cho chúng ta một cái nhìn tổng quan về tiềm năng, tính hiệu quả, khả năng mở rộng và ứng dụng vào thực tế của việc áp dụng VR/ AR vào công nghiệp cho quá trình tự động hóa.

4.1. KẾT QUẢ ĐẠT ĐƯỢC

- Hiển thị hình ảnh từ camera lên ứng dụng VR: Nhóm đã hiện thực thành công lấy dữ liệu từ ba Camera IP và hiển thị đồng thời lên ứng dụng.
- Điều khiển camera thông qua kính VR: Nhóm đã hiện thực thành công chức năng xoay đầu khi đeo kính VR để cho camera Hikvision xoay theo, đây là một chức năng quan trọng để giúp người dùng có được một cái nhìn tổng quan về chỗ làm việc.
- Hiển thị thông số của Robot trong môi trường VR: Nhóm đã hiện thực thành công MQTT Client đơn giản chứa các method để lấy các thông số của cánh tay (tọa độ, tốc độ, gốc servo,...) và các method

phục vụ cho việc điều khiển cánh tay. Các method này sẽ gọi tới các ROS service được hiện thực trong chương trình điều khiển cánh tay.

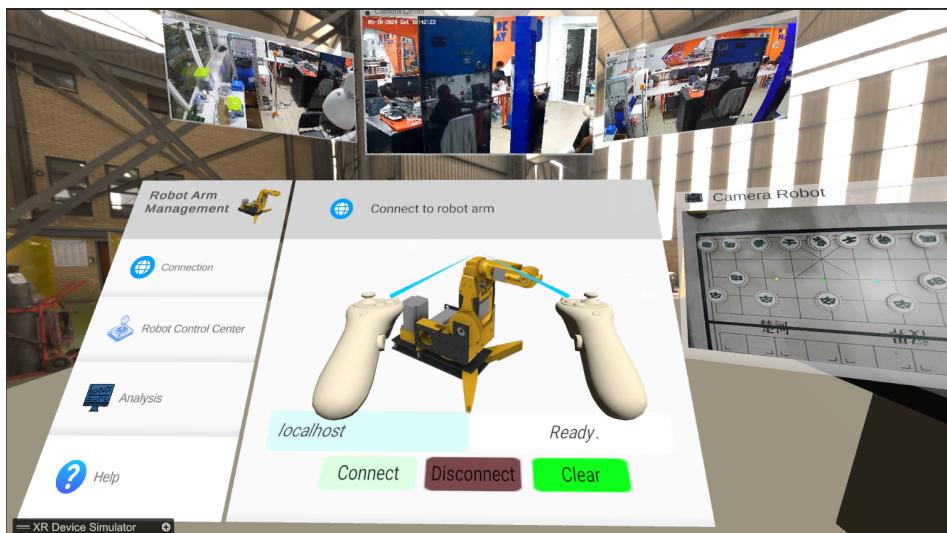
- Đồng bộ giữa chuyển động của robot ảo và robot thật: bằng cách nhận các dữ liệu về góc xoay từ robot thật, xử lý và thực hiện xoay các khớp tương ứng cho robot ảo với độ trễ khoảng giây.

4.1.1. VR APPLICATION

4

Ứng dụng VR được nhóm thực hiện như một ứng dụng digital twin nhằm điều khiển cánh tay Robot. Ứng dụng được phát triển sử dụng Unity, sinh ra file apk và nạp, sử dụng trên kính Oculus Quest 2. Khi vào ứng dụng, ta sẽ thấy được khung cảnh VR, mô hình 3D của cánh tay, và một bảng menu để điều khiển và 3 camera hiển thị khung cảnh nơi làm việc.

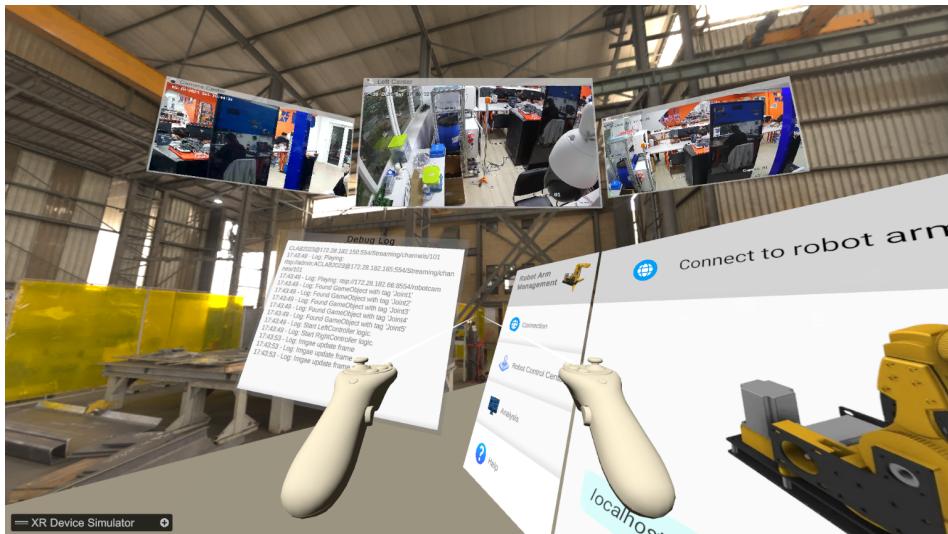
Ba camera được đặt ở phía trên bảng điều khiển và có thể phóng to hỗ trợ quan sát chi tiết.



Hình 4.1: Giao diện ứng dụng khi vừa khởi động

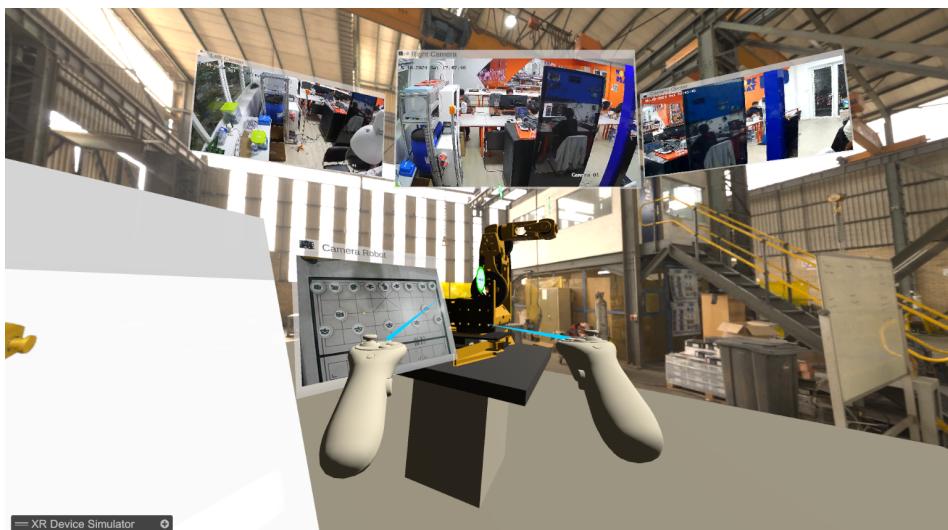
Khi vừa khởi động, ta có thể thấy cụm camera ở giữa, có thể nhấn vào để phóng to, một bảng điều khiển để người dùng có thể tương tác được với các chức năng của cánh tay máy Robot. Bên phải sẽ là mô hình 3D,

mô hình này sẽ ánh xạ chuyển động đối với mô hình thật và bên phải sẽ là khung cửa sổ ghi lại các nhật trong unity, khung này có tác dụng giúp ta sửa lỗi (debug) khi đang chạy chương trình.



4

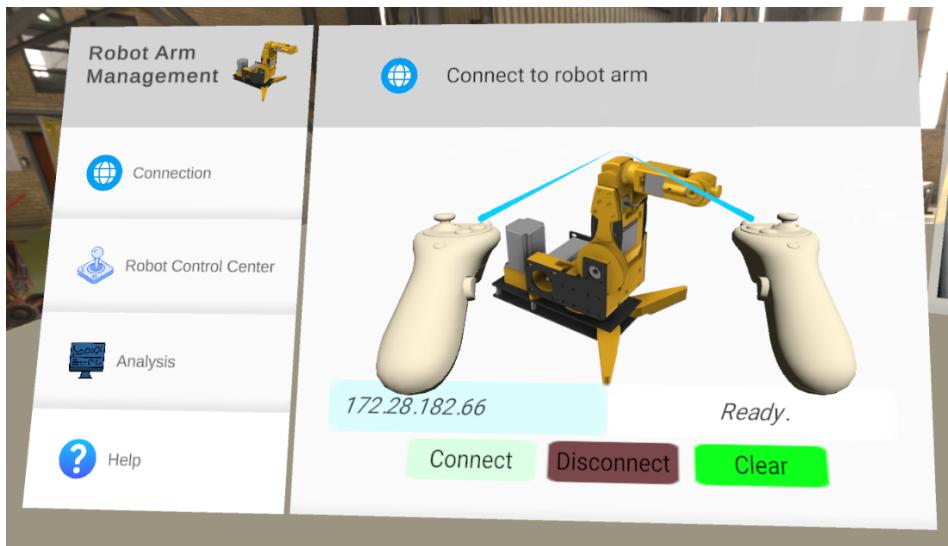
Hình 4.2: Giao diện ứng dụng bên góc trái của người sử dụng hiển thị debug log trong chương trình unity



Hình 4.3: Giao diện ứng dụng bên góc phải của người sử dụng hiển thị camera của robot và cánh tay robot ảo

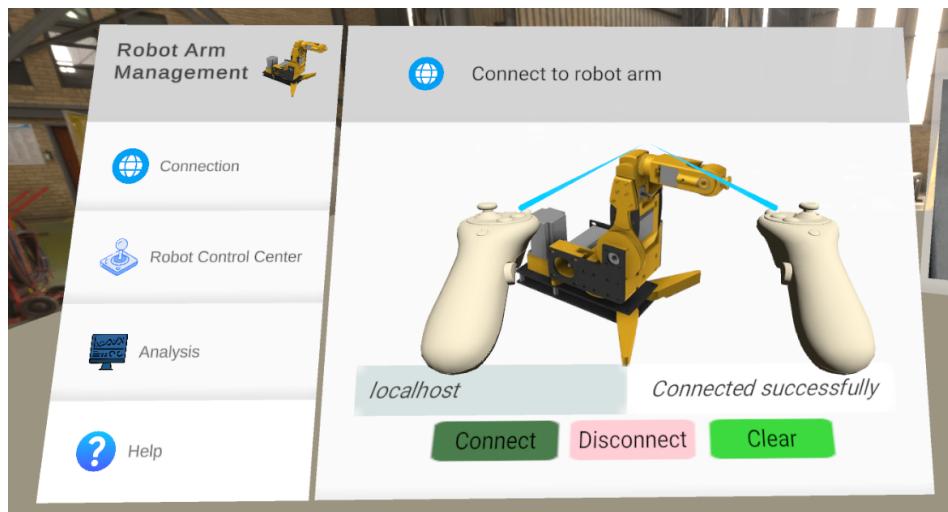
Trong bảng điều khiển cánh tay robot, khi mới bắt đầu vào sẽ ở trong mục "Connection". Người dùng có thể nhập địa chỉ IP của robot hoặc để giá trị mặc định và nhấn nút "Connect" để kết nối tới cánh tay robot. Nút "Disconnect" bị vô hiệu hóa do chưa kết nối tới cánh tay robot.

4



Hình 4.4: Giao diện ứng dụng khi ở mục Connection

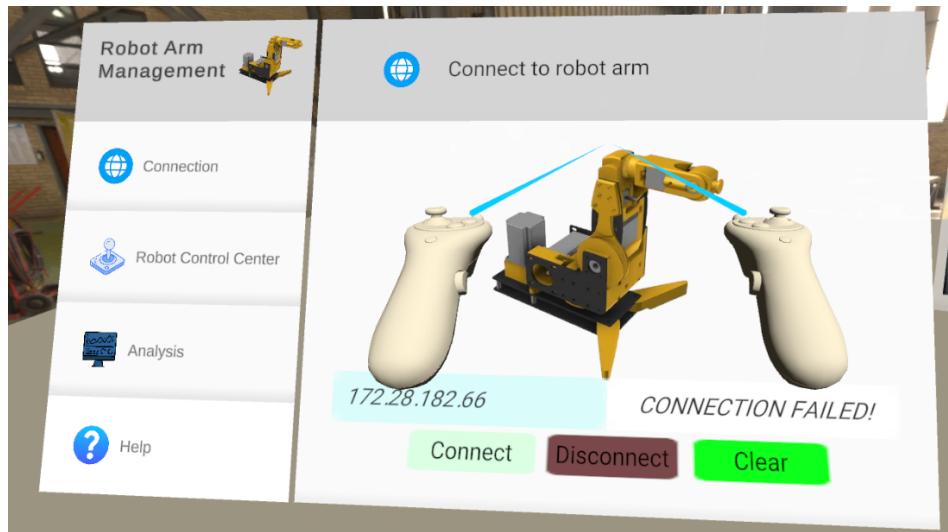
Khi kết nối thành công hiện thông báo "Connection Successfully". Nút "Disconnect" được bật và nút "Connect" bị vô hiệu hóa



4

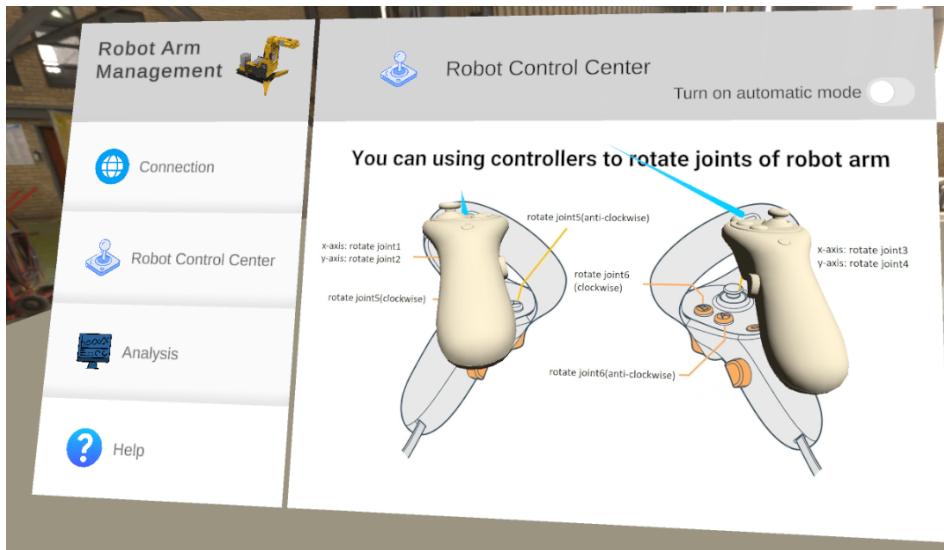
Hình 4.5: Giao diện khi kết nối thành công với cánh tay robot

Khi kết nối thất bại hiện thông báo "Connection FAILED!".



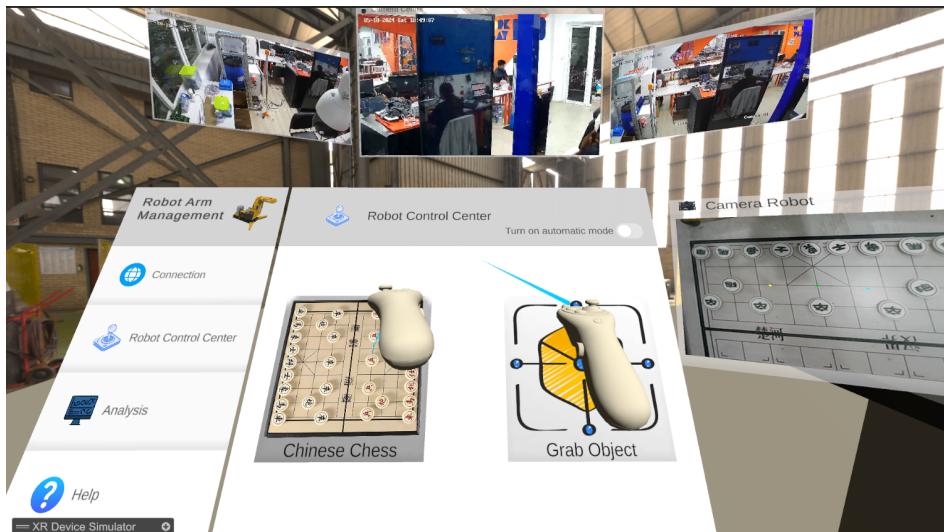
Hình 4.6: Giao diện khi kết nối thất bại với cánh tay robot

Tiếp đến mục "Robot Control Center", ban đầu điều khiển ở manul mode. Người dùng nhấn các nút trên tay cầm điều khiển để xoay các khớp của cánh tay robot.



Hình 4.7: Giao diện bắt đầu của bảng điều khiển với manaul mode

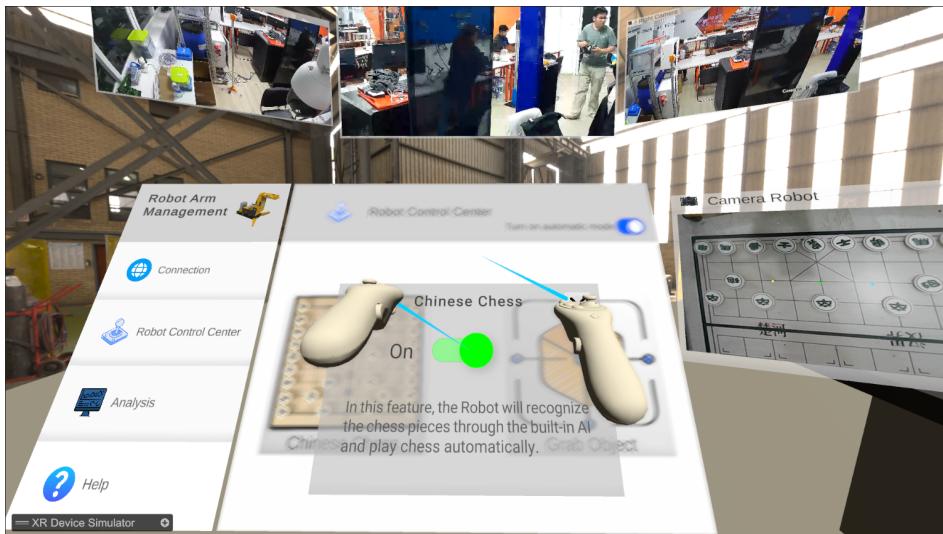
Người dùng có thể nhấn vào nút "Turn on automatic mode" ở trên bên phải để chuyển qua chế độ Auto mode. Có thể chuyển đổi qua lại giữa hai chế độ.



Hình 4.8: Giao diện của bảng điều khiển với Auto mode

Khi nhấn vào một trong các chức năng sẽ mở ra menu con để bật tắt

hoặc có các tùy chọn thêm cho tính năng. Người dùng có thể bật tính năng chơi cờ tướng tự động khi nhấn vào nút "On".

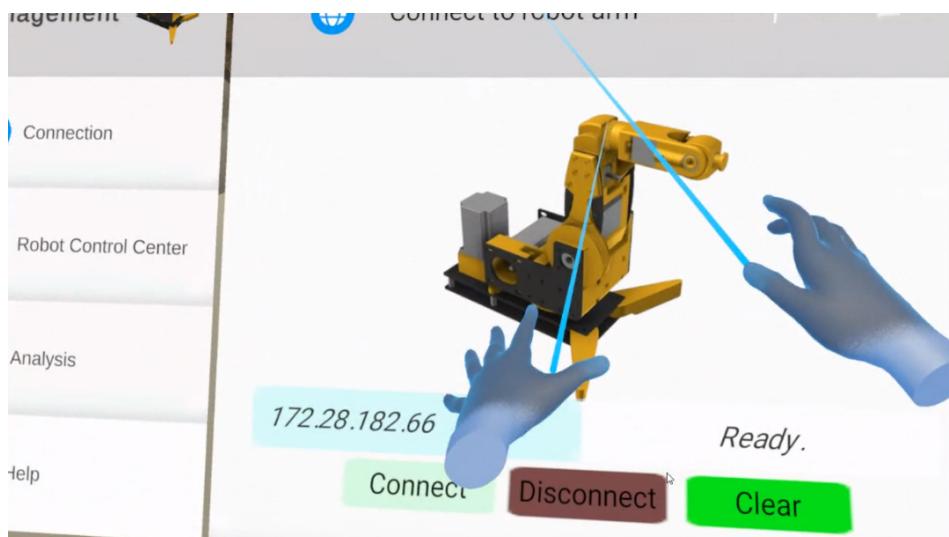


Hình 4.9: Giao diện Menu con của chức năng Chinese Chess

Nhận dạng bàn tay và tương tác với bảng điều khiển

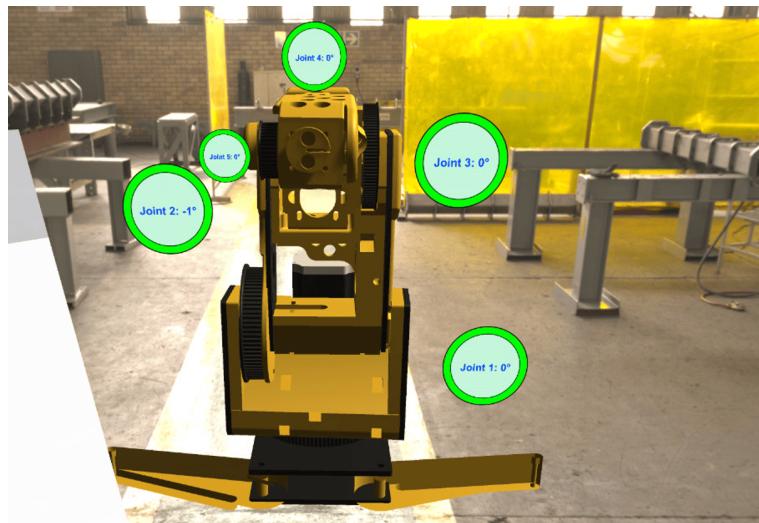
Ứng dụng VR của nhóm đã nhận dạng được cánh tay [44], và có thể tương tác với bảng điều khiển mức cơ bản. Việc nhận dạng cử chỉ như nhấp, kéo, thả còn ở mức hạn chế. Và cũng như liên quan đến độ chính xác thực hiện điều khiển robot thực tế, nhóm nên vẫn ưu tiên sử dụng tay cầm của Kính VR. Tích hợp được chức năng này tăng trải nghiệm người dùng khi sử dụng ứng dụng và là bước nền tảng hiện thực thêm chức năng tương tác với bàn tay sau này.

4



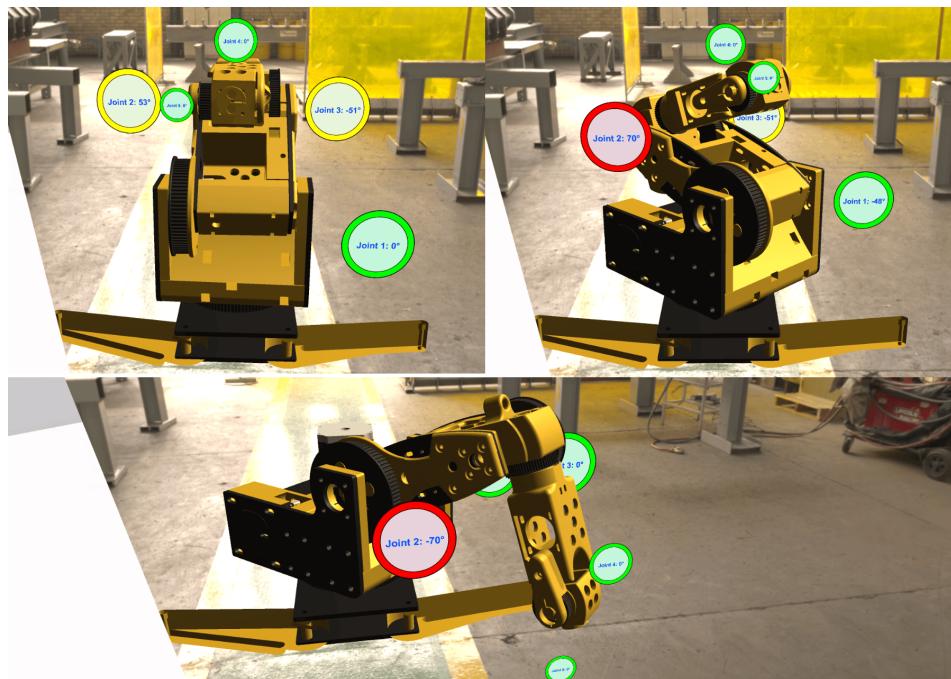
Hình 4.10: Nhận dạng bàn tay và hiển thị lên kính VR

Như đã đề cập, Robot sẽ ghi nhận các thông số về góc quay của các khớp. Khi giá trị này đạt tới ngưỡng giới hạn, hiển thị màu vàng; và khi vượt quá ngưỡng giới hạn, hiển thị màu đỏ. Nếu giá trị không nằm trong hai khoảng đó, thì hiển thị màu xanh, giúp người điều khiển dễ dàng nhận biết tình trạng góc quay của Robot. Dưới đây là một vài hình ảnh hiển thị thông số của các khớp trong trường hợp khác nhau.

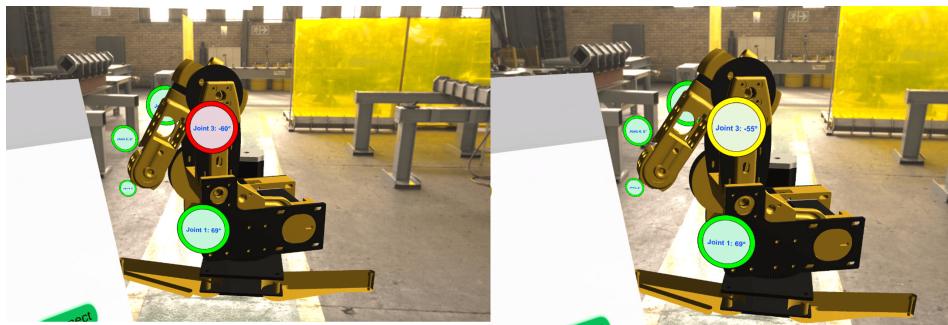


4

Hình 4.11: Hình ảnh thông số ở trạng thái mặc định, không có khớp nào được quay



Hình 4.12: Hình ảnh thông số của khớp thứ 2 ở hai trường hợp gần tới ngưỡng và đạt ngưỡng giới hạn



Hình 4.13: Hình ảnh thông số của khớp thứ 3 ở hai trường hợp gần tới ngưỡng và đạt ngưỡng giới hạn

4

4.2. KIỂM TRA ĐÁNH GIÁ HỆ THỐNG

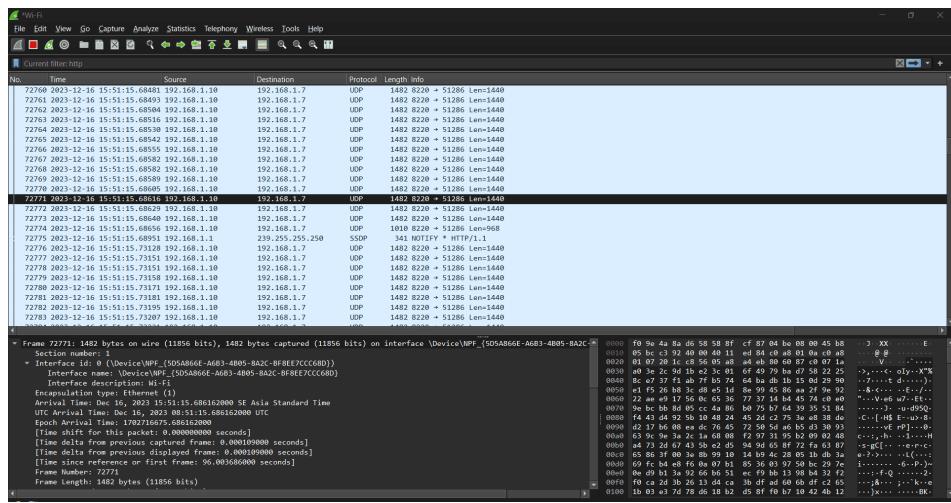
Ở phần này, nhóm sẽ thực hiện một số thử nghiệm để đánh giá hoạt động và khả năng của hệ thống. Nhóm sẽ đánh giá về mức độ sử dụng CPU, GPU và số lượng khung ảnh trên giây (Frame per second - fps) của ứng dụng trên kính Oculus. Ngoài ra nhóm sẽ đánh giá thêm về độ trễ của giao thức MQTT, độ trễ giữa robot ảo và robot thật.

4.2.1. VỀ THỜI GIAN CẬP NHẬT MỖI GÓI TIN TRONG TRÊN MỖI FRAME ẢNH TRONG VIỆC STREAMING

Dưới đây là thời gian đến của mỗi gói tin trong việc update frame cho việc streaming Video:

Ở đây ta sẽ thực hiện lấy thời gian đến (arrival_time) của 11 gói tin từ **No.72760** đến **No.72770** để thực hiện tính toán về chênh lệch thời gian đến trung bình của các gói tin là khoảng bao nhiêu. Ta có thời gian đến trung bình của các gói tin lần lượt là:

- **No 72760:** Dec 16, 2023 15:51:15.684815000 SE Asia Standard Time
- **No 72761:** Dec 16, 2023 15:51:15.684933000 SE Asia Standard Time
- **No 72762:** Dec 16, 2023 15:51:15.685044000 SE Asia Standard Time
- **No 73763:** Dec 16, 2023 15:51:15.685160000 SE Asia Standard Time



Hình 4.14: Các gói tin cập nhật frame

No.	Time	Source	Destination	Protocol	Length	Info
72760	2023-12-16 15:51:15.684993	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72761	2023-12-16 15:51:15.684993	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72762	2023-12-16 15:51:15.685094	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72763	2023-12-16 15:51:15.685116	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72764	2023-12-16 15:51:15.68530	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72765	2023-12-16 15:51:15.68542	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72766	2023-12-16 15:51:15.68555	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72767	2023-12-16 15:51:15.68582	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72768	2023-12-16 15:51:15.68582	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72769	2023-12-16 15:51:15.68589	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72770	2023-12-16 15:51:15.68605	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72771	2023-12-16 15:51:15.68616	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72772	2023-12-16 15:51:15.68629	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72773	2023-12-16 15:51:15.68640	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72774	2023-12-16 15:51:15.68656	192.168.1.10	192.168.1.7	UDP	1010	8220 → 51286 Len=968
72775	2023-12-16 15:51:15.68951	192.168.1.1	239.255.255.250	SSDP	341	NOTIFY * HTTP/1.1
72776	2023-12-16 15:51:15.73128	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72777	2023-12-16 15:51:15.73151	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72778	2023-12-16 15:51:15.73151	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72779	2023-12-16 15:51:15.73158	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72780	2023-12-16 15:51:15.73171	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72781	2023-12-16 15:51:15.73181	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72782	2023-12-16 15:51:15.73195	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72783	2023-12-16 15:51:15.73207	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440
72784	2023-12-16 15:51:15.73207	192.168.1.10	192.168.1.7	UDP	1482	8220 → 51286 Len=1440

Hình 4.15: Các gói tin cập nhật frame - tiếp theo

- **No 73764:** Dec 16, 2023 15:51:15.685305000 SE Asia Standard Time
- **No 73765:** Dec 16, 2023 15:51:15.685422000 SE Asia Standard Time
- **No 73766:** Dec 16, 2023 15:51:15.685554000 SE Asia Standard Time
- **No 73767:** Dec 16, 2023 15:51:15.685824000 SE Asia Standard Time
- **No 73768:** Dec 16, 2023 15:51:15.685824000 SE Asia Standard Time

- **No 73769:** Dec 16, 2023 15:51:15.685898000 SE Asia Standard Time
- **No 73770:** Dec 16, 2023 15:51:15.686053000 SE Asia Standard Time

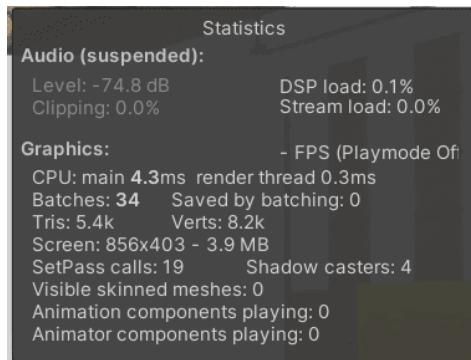
Thời gian chênh lệch giữa các khoảng tính từ trên xuống dưới lần lượt là:

1. $\text{DifTime} = \text{ArrivalTime}_{\text{No.73761}} - \text{ArrivalTime}_{\text{No.73760}} = 0.118 \text{ ms}$
2. $\text{DifTime} = \text{ArrivalTime}_{\text{No.73762}} - \text{ArrivalTime}_{\text{No.73761}} = 0.111 \text{ ms}$
3. $\text{DifTime} = \text{ArrivalTime}_{\text{No.73763}} - \text{ArrivalTime}_{\text{No.73762}} = 0.116 \text{ ms}$
4. $\text{DifTime} = \text{ArrivalTime}_{\text{No.73764}} - \text{ArrivalTime}_{\text{No.73763}} = 0.145 \text{ ms}$
5. $\text{DifTime} = \text{ArrivalTime}_{\text{No.73765}} - \text{ArrivalTime}_{\text{No.73764}} = 0.117 \text{ ms}$
6. $\text{DifTime} = \text{ArrivalTime}_{\text{No.73766}} - \text{ArrivalTime}_{\text{No.73765}} = 0.132 \text{ ms}$
7. $\text{DifTime} = \text{ArrivalTime}_{\text{No.73767}} - \text{ArrivalTime}_{\text{No.73766}} = 0.280 \text{ ms}$
8. $\text{DifTime} = \text{ArrivalTime}_{\text{No.73768}} - \text{ArrivalTime}_{\text{No.73767}} = 0.0 \text{ ms}$
9. $\text{DifTime} = \text{ArrivalTime}_{\text{No.73769}} - \text{ArrivalTime}_{\text{No.73768}} = 0.74 \text{ ms}$
10. $\text{DifTime} = \text{ArrivalTime}_{\text{No.73770}} - \text{ArrivalTime}_{\text{No.73769}} = 0.155 \text{ ms}$

Vậy thời gian chênh lệch trung bình giữa các gói tin sau khi gửi sẽ ở khoảng: **0.1248 ms** với giá trị nhỏ nhất trong thời gian chênh lệch hiện tại là **0.0 ms** và lớn nhất dựa trên số liệu trên là **0.280 ms**

4.2.2. VỀ SỰ THAY ĐỔI VỀ THÔNG SỐ TRONG UNITY

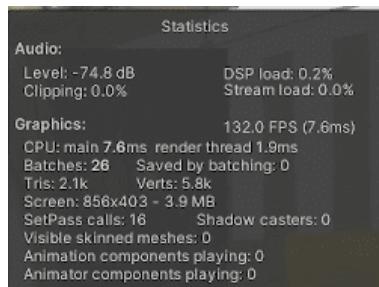
Khi chưa hiện Run trong Unity, dưới đây là hình ảnh biểu diễn về các thông số Statics:



Hình 4.16: Hình ảnh biểu diễn trạng thái Static khi chưa Run trong Unity

4

Sau khi đã thực hiện nhấn Run thì chỉ số Graphics của Statistics đã có sự thay đổi.



Hình 4.17: Hình ảnh biểu diễn trạng thái Static khi đã Run trong Unity

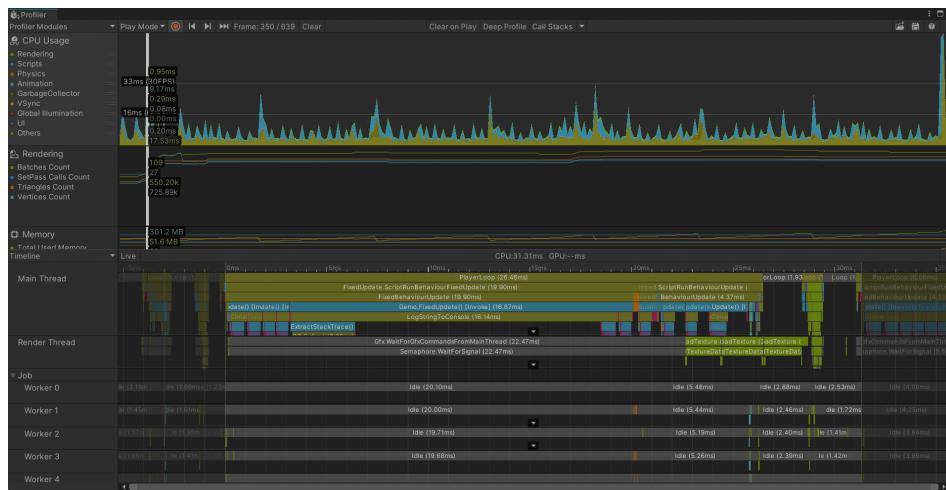
Ta nhận thấy được sự thay đổi của thông số FPS trong Unity cập nhật trung bình khoảng 132FPS (7.6ms)

Đây là số liệu về việc Render Graphic (Các phần tử UI trong Unity - không liên quan đến thông số FPS trong việc render Video)

Dưới đây là dữ liệu ghi lại của Profiler trong Unity:

Trường hợp 1:

Dựa vào những Record ta nhận thấy trong MainThread của Unity lúc rendering UI CPU chiếm dụng là khoảng **31.31 ms** và trong lúc bình thường không rendering là khoảng **6.4 ms**, về GPU thì hiện tại đang không

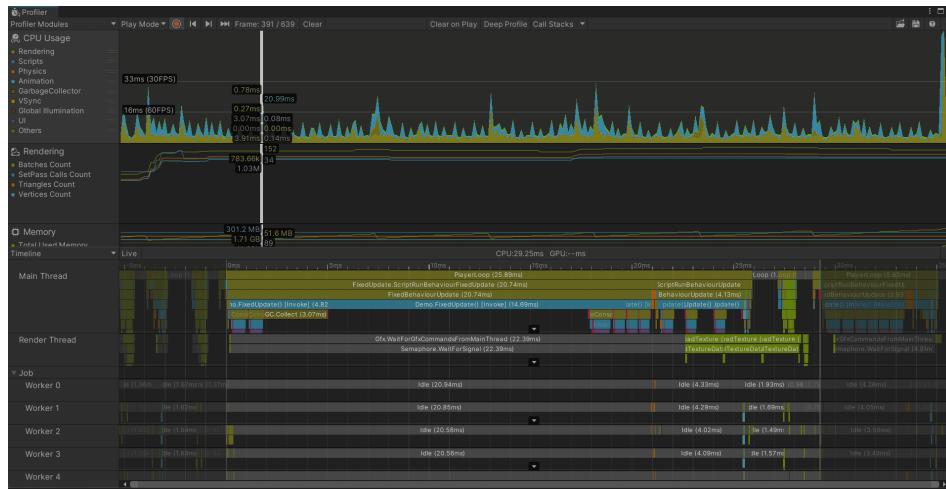


4

Hình 4.18: Phân tích các thành phần trong Unity

chiếm dụng, trong khi đó idle time giữa các lần làm việc trung bình là khoảng **5.41 ms**. Tổng thời gian quá trình đợi Render từ lần trước đến lần tiếp theo của trường hợp này bằng **$31.31 + 6.4 + 5.41 = 43.12 \text{ ms}$**

Trường hợp 2:

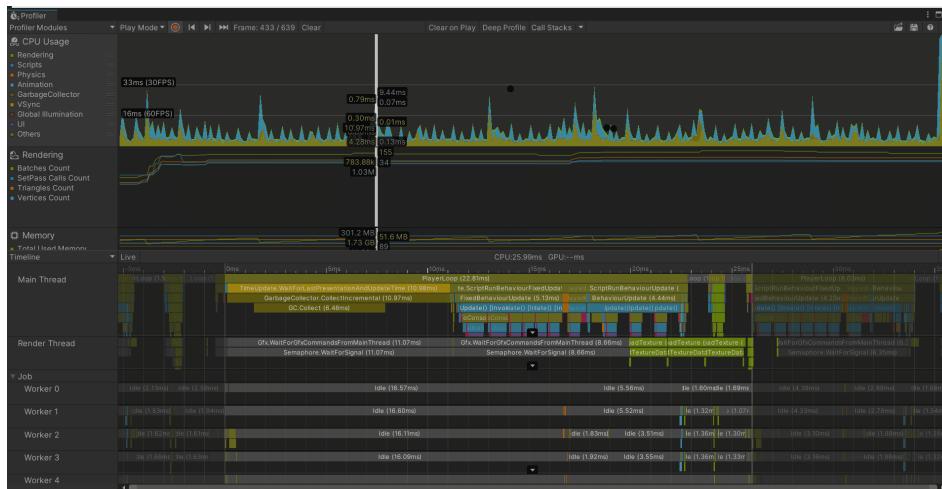


Hình 4.19: Phân tích các thành phần trong Unity

Dựa vào những Record ta nhận thấy trong MainThread của Unity

lúc rendering UI CPU chiếm dụng là khoảng **29.25ms** và trong lúc bình thường không rendering là khoảng **5.02 ms**, về GPU thì hiện tại đang không chiếm dụng, trong khi đó idle time giữa các lần làm việc trung bình là khoảng **6.26 ms**. Tổng thời gian quá trình đợi Render từ lần trước đến lần tiếp theo của trường hợp này bằng $29.25 + 5.02 + 6.26 = 40.53 \text{ ms}$

Trường hợp 3:



4

Hình 4.20: Phân tích các thành phần trong Unity

Dựa vào những Record ta nhận thấy trong MainThread của Unity lúc rendering UI CPU chiếm dụng là khoảng **25.99ms** và trong lúc bình thường không rendering là khoảng **3.29 ms**, về GPU thì hiện tại đang không chiếm dụng, trong khi đó idle time giữa các lần làm việc trung bình là khoảng **3.54 ms**. Tổng thời gian quá trình đợi Render từ lần trước đến lần tiếp theo của trường hợp này bằng $25.99 + 3.29 + 3.54 = 32.82 \text{ ms}$

Kiểm tra trên code VLC Streaming trên C# Dưới đây là code C# đã thêm việc kiểm tra thời gian mỗi lần khung hình được Update:

Ý tưởng đơn giản là ta sẽ xem timestamp khi frame được cập nhật, từ đó ta sẽ biết được khoảng thời gian giữa 2 lần liên tiếp frame được cập nhật, ta sẽ tính được số khung cảnh trên 1s của video stream

Kết quả được in ra trong **Debug.Log** trên Unity như sau:



Hình 4.21: Log thể hiện thời gian update mỗi frame

4

Dựa vào log trên ta thấy rằng thời gian trung bình của mỗi lần update được ghi ra cuối cùng bằng **0.04247347** xấp xỉ khoảng **42ms**. Thực hiện so sánh với kết quả được đánh giá trong Profiler của Unity (khoảng thời gian mỗi lần UI Rendering toàn bộ - có chứa UI Video) thì khoảng thời gian trung bình là **42ms** sẽ cho ra FPS ổn định ở mức độ là khoảng 25 khung hình trên giây tương đương với **25 FPS**.

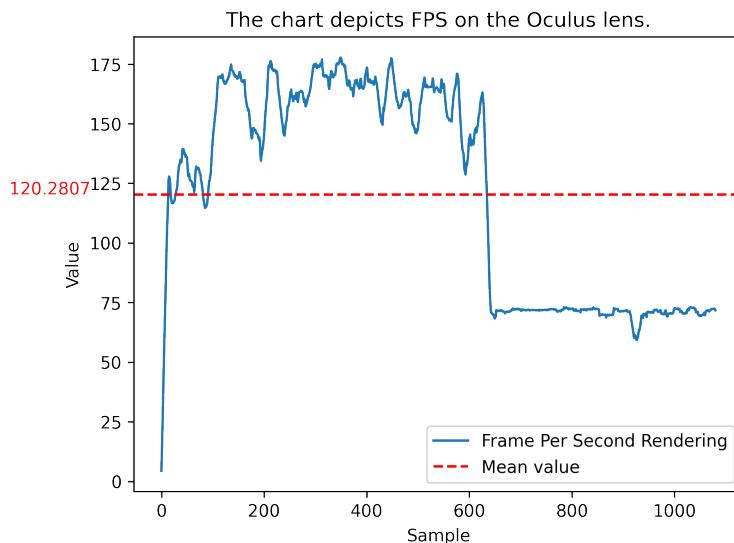
Với thời gian mỗi gói tin được cập nhật từ request http tối đa là 0.280ms thì chênh lệch giữa tốc độ render ra mỗi frame là không đáng kể quá nhiều, thay vào đó FPS có thể lên đến tối đa là **180FPS** trong một vài trường hợp có lịch sử chênh lệch thời gian là **0.0055s** và nhỏ nhất khoảng **10 FPS** với sự chênh lệch thời gian khoảng **0.09s**.

4.2.3. SỐ KHUNG HÌNH TRÊN GIÂY CỦA ỨNG DỤNG OCULUS

Nhóm sẽ đo số khung hình trên giây (FPS) khi không đeo kính Oculus vào và khi bắt đầu đeo kính, tương tác với thế giới ảo với số lượng mẫu là 1080 đo trong khoảng 2 phút. Dưới đây ta có thể thấy bắt đầu FPS sẽ tăng dần đến khoảng 175 FPS, khi đeo vào thì FPS sẽ giảm dần đến 71 FPS và ổn định ở đó.

Có thể thấy rằng với mức FPS khoảng 71, người dùng sẽ có trải nghiệm tốt mà không bị giật lag. Để đảm bảo trải nghiệm mượt mà và thoải mái trong môi trường thực tế ảo (VR), các chuyên gia thường khuyến cáo mức FPS tối thiểu là 60 FPS. Mức này được coi là đủ để giảm thiểu hiện tượng nhức đầu, chóng mặt và cảm giác khó chịu do giật lag khi tương tác trong thế giới ảo nói riêng và video game nói chung [11]. Với 71 FPS, ứng dụng

không chỉ đáp ứng mà còn vượt qua ngưỡng tối thiểu này, đảm bảo rằng người dùng sẽ có trải nghiệm thị giác ổn định và mượt mà.



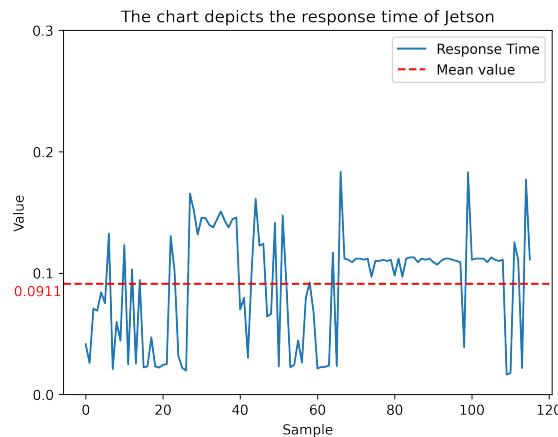
4

Hình 4.22: Số khung hình trên giây (FPS) khi không đeo kính và khi đeo kính Oculus

4.2.4. VỀ THỜI GIAN TRỄ CỦA GIAO THỨC MQTT

Khi người dùng điều khiển robot bằng joystick đến khi robot nhận được dữ liệu thông qua giao thức MQTT, với số lượng mẫu đo là 110, thời gian phản hồi tối thiểu là 0.02189s, tối đa là 0.183s và thời gian trung bình đo được là 0.0911. Với khoảng thời gian phản hồi như vậy, hệ thống có khả năng cung cấp một trải nghiệm điều khiển gần như thời gian thực, đảm bảo sự mượt mà và hiệu quả trong việc điều khiển robot. Thời gian phản hồi ngắn giúp giảm độ trễ trong quá trình điều khiển, từ đó tăng cường khả năng tương tác và phản ứng nhanh chóng của robot theo các lệnh từ người điều khiển.

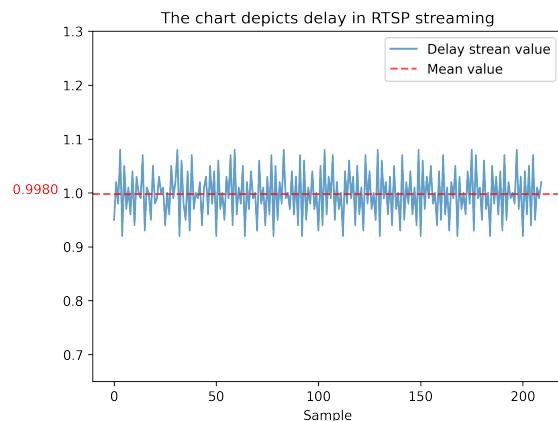
4



Hình 4.23: Độ trễ của giao thức MQTT

4.2.5. VỀ THỜI GIAN TRỄ KHUNG HÌNH KHI STREAM RTSP

Khi hình ảnh từ camera IP được lấy về, thường sẽ có một độ trễ nhất định trong quá trình truyền tải. Nhóm sẽ tiến hành đo độ trễ này bằng cách thu thập mẫu dữ liệu từ 210 lần truyền tải khác nhau. Kết quả cho thấy rằng độ trễ nhỏ nhất (min delay) là 0.92 giây và độ trễ lớn nhất là 1.08 giây, với giá trị trung bình là 0.998 giây.



Hình 4.24: Độ trễ khi lấy luồng video bằng RTSP trên unity

5

TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

5.1. TỔNG KẾT

Tổng kết lại quá trình phát triển và hiện thực hệ thống, nhóm đã đạt được một số kết quả tiêu biểu như sau:

- Về kiến trúc: Nhóm đã hiện thực thành công hệ thống dựa trên mô hình kiến trúc mà nhóm đã đề xuất ở chương 3, với 3 lớp chính là lớp ảo (virtual), lớp mạng (network) và lớp vật lý (physical).
- Về chức năng điều khiển cánh tay Robot và IPCamera từ môi trường ảo VR: Ứng dụng cho phép người dùng dễ dàng kiểm soát cánh tay Robot thông qua các thao tác đơn giản trên ứng dụng VR. Đồng thời, người dùng có thể theo dõi trạng thái của cánh tay qua mô hình 3D ảo, cũng như xoay đầu trong không gian ảo để điều khiển IPCamera xoay theo, mở ra khả năng quan sát toàn diện của môi trường làm việc.
- Nhóm đã thành công trong việc tạo ra một bảo sao ảo (Digital Twin) cho cánh tay robot bằng cách tích hợp mô phỏng chuyển động của

cánh tay thực tế vào môi trường ảo. Đồng thời kết nối với camera được gắn ở robot để quan sát thao tác trực tiếp trên cánh tay.

- Nhóm đã hiện thực thành công chức năng quay kính Oculus và lấy góc nhìn của camera tương ứng đến vị trí xoay đầu của người đeo kính.
- Về VR app chạy trên thiết bị thực tế ảo Oculus Quest2: Nhóm hiện thực thành công một mô hình 3d của cánh tay, đây là một bản sao hoạt động đồng bộ với cánh tay được phát triển trên Unity3D, cùng với việc hiển thị một vài thông số của cơ bản của Robot như giới hạn góc xoay,... trong môi trường VR, điều đó giúp mang lại một trải nghiệm mới mẻ cho người dùng và cũng phần nào minh chứng được khả năng làm chủ những công nghệ mới của thế hệ sinh viên Việt Nam ngày nay.

5

Tuy nhiên, do hạn chế về mặt thiết bị, thời gian và nguồn lực nên hệ thống của nhóm dù hoạt động tốt những vẫn còn tồn tại một số điểm hạn chế như sau:

- Sử dụng Jetson Nano như một gateway không được coi là lựa chọn tối ưu, bởi vì dù có hiệu suất tính toán tốt, nhưng nó có một số hạn chế quan trọng. Jetson Nano được thiết kế chủ yếu để xử lý công việc máy học và thị giác máy tính, không phải làm gateway chuyên sâu.
- Mô hình 3D trong VR app con tương đối đơn giản, chưa có nhiều khả năng về mặt mô phỏng.
- Việc phát triển và tích hợp phần mềm còn hạn chế về sự hỗ trợ của các phiên bản có thể được hỗ trợ ở cả phần cứng lẫn phần mềm, khiến lựa chọn về việc sử dụng công nghệ của nhóm còn chưa tối ưu về mặt hiệu năng.
- Còn ít trải nghiệm để có thể đánh giá được tiện ích của nó mang lại trong thực tế.

5.2. HƯỚNG PHÁT TRIỂN TƯƠNG LAI

Để cải thiện hệ thống trong tương lai, nhóm sẽ cân nhắc các giải pháp như sau:

- Tìm hiểu và sử dụng OPC UA thay vì MQTT để trao đổi dữ liệu giữa các lớp vật lý và lớp ảo thông qua lớp mạng. OPC UA mang lại các ưu điểm như khả năng tương thích cao, tính bảo mật mạnh mẽ và kiến trúc linh hoạt, giúp tăng cường tính linh hoạt và độ tin cậy của hệ thống trao đổi dữ liệu.
- Nhóm sẽ đánh giá và sử dụng Raspberry Pi làm gateway trong tương lai hoặc bất cứ ai có hứng thú về đề tài của chúng tôi thì có thể tham khảo hướng phát triển này. Raspberry Pi cung cấp một loạt các phiên bản với nhiều tùy chọn cổng kết nối và hỗ trợ mạng Gigabit Ethernet, giúp cải thiện khả năng truyền dữ liệu. Đặc biệt, Raspberry Pi thường dễ dàng mở rộng và có cộng đồng sử dụng lớn, giúp đơn giản hóa quá trình phát triển và xử lý các vấn đề liên quan đến việc sử dụng nó làm gateway.
- Hiện tại, máy tính nhúng (SBC - single board computer) nhóm đang xài là Jetson Nano, và hệ điều hành là Ubuntu 16.04. Tuy nhiên khi xài hệ điều hành từ nhà phát triển có sẵn thì có rất nhiều những package mà chúng ta không cần đến và điều này ảnh hưởng đến hiệu năng của hệ thống. Nhóm rất muốn tự mình hiệu chỉnh lại một linux kernel cho SBC bằng Yocto, chỉ cài những package cần thiết và lược bỏ các packages/services nhiều nhất có thể. Điều này chắc chắn sẽ tăng tốc độ boot và hiệu năng của máy tính nhúng. Tuy nhiên do hạn chế về mặt thời gian, nhóm chưa thể hiện thực được ý tưởng này.
- Tăng tính tương đồng và phức tạp của mô hình 3D trong VR app để đảm bảo tính chân thực và mô phỏng được các hoạt động và hành vi của cánh tay.
- Đề xuất sử dụng công nghệ mới hơn ở lớp Physical để có thể dễ dàng tích hợp các tính năng nâng cao và phức tạp hơn.
- Hiện tại Robot chỉ đang tự đánh cờ, chúng ta có thể thêm một vài thứ cho Robot như gắn thêm cảm biến để theo dõi vị trí của robot trong nhà máy và thu thập dữ liệu về các thao tác vận hành. Robot

đã có sẵn camera của nó và hình ảnh 3 camera ở trong nhà máy, được gắn ở phần đầu cánh tay, ta có thể thêm tính năng tự động gửi cảnh báo hoặc báo cáo cho nhân viên quản lý khi phát hiện sự cố,...

TÀI LIỆU THAM KHẢO

- [1] “05 blockly-color sort.” (), [Online]. Available: <https://chrisandjimcim.com/5-blockly-color-sort/>.
- [2] A. Ahad, Z. R. Khan, and S. A. Ahmad, “Intelligent parking system,” *World Journal of Engineering and Technology*, vol. 4, no. 2, pp. 106–113, 2016. DOI: 10 . 4236 / wjet . 2016 . 42014. [Online]. Available: <http://www.scirp.org/journal/wjet>.
- [3] “Ai robot - object tracking | object following robot using tensorflow lite.” (), [Online]. Available: <https://helloworld.co.in/article/ai-robot-object-tracking-object-following-robot-using-tensorflow-lite>.
- [4] T. Alam, “A reliable communication framework and its use in internet of things (iot),” *CSEIT1835111*, vol. Received 10, pp. 450–456, 2018.
- [5] N. T. Ân, P. T. Danh, H. N. Hùng, *et al.*, *Phát triển IoT Gateway bằng Python*.
- [6] “An overview of http.” (), [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>.
- [7] “Cánh tay robot công nghiệp và những ứng dụng trong sản xuất.” (), [Online]. Available: <https://itgtechnology.vn/canh-tay-robot-cong-nghiep>.
- [8] S. Casey Baseel. “Remote-control vr robots to start working in japanese convenience stores this summer.” (), [Online]. Available: <https://japantoday.com/category/tech/remote-control-vr-robots-to-start-working-in-japanese-convenience-stores-this-summer>.
- [9] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, “Smart factory of industry 4.0: Key technologies, application case, and challenges,” *IEEE Access*, vol. 6, pp. 6505–6519, 2018. DOI: 10 . 1109 / ACCESS . 2017 . 2783682.

- [10] Y. Chen, “Integrated and intelligent manufacturing: Perspectives and enablers,” *Engineering*, vol. 3, no. 5, pp. 588–595, 2017.
- [11] M. Claypool, K. Claypool, and F. Dama, “The effects of frame rate and resolution on users playing first person shooter games,” 2024.
- [12] “Configure unity settings.” (), [Online]. Available: <https://developer.oculus.com/documentation/unity/unity-conf-settings/>.
- [13] “Dotween (hotween v2).” (), [Online]. Available: <https://assetstore.unity.com/packages/tools/animation/dotween-hotween-v2-27676>.
- [14] F. Dunn and I. Parberry, *3D Math Primer for Graphics and Game Development* (2nd Ed).
- [15] A. Fuller, Z. Fan, C. Day, and C. Barlow, “Digital twin: Enabling technologies, challenges and open research,” *IEEE access*, vol. 8, pp. 108952–108971, 2020.
- [16] E. Glaessgen and D. Stargel, “The digital twin paradigm for future nasa and us air force vehicles,” in *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, 2012, p. 1818.
- [17] “H. 264 encoder using gstreamer,” in *2015 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Mar. 2015. DOI: 10.1109/ICCPCT.2015.7159511.
- [18] R. L. H. Schulzrinne A. Rao, “Rfc2326: Real time streaming protocol (rtsp),” *RFC Editor, United States*, 1998. DOI: <https://doi.org/10.17487/RFC2326>.
- [19] A. Haghshenas, A. Hasan, O. Osen, and E. T. Mikalsen, “Predictive digital twin for offshore wind farms,” *Energy Informatics*, vol. 6, no. 1, pp. 1–26, 2023.
- [20] “Hands interaction demo.” (), [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.5/manual/samples-hands-interaction-demo.html>.
- [21] L. Hikvision Digital Technology Co., *IP Surveillance API PTZ Service Specification*. 2013, pp. 13–14. [Online]. Available: https://download.catalogosicurezza.com/DOWNLOAD/Hikvision/Software/Pacchetti%20per%20Sviluppo/05%20%20%20ISAPI/HIKVISION%20ISAPI_2.0-PTZ%20Service.pdf.

- [22] “How do i get my rtsp stream?” (), [Online]. Available: <https://supportusa.hikvision.com/support/solutions/articles/17000129064-how-do-i-get-my-rtsp-stream-%20>.
- [23] “Interaction sdk.” (), [Online]. Available: <https://developer.oculus.com/documentation/unity/unity-isdk-interaction-sdk-overview/>.
- [24] “Libvlc player for unity.” (), [Online]. Available: <https://github.com/bosqmode/UnityVLCPlayer>.
- [25] M. Lohtander, N. Ahonen, M. Lanz, J. Ratava, and J. Kaakkunen, “Micro manufacturing unit and the corresponding 3d-model for the digital twin,” *Procedia manufacturing*, vol. 25, pp. 55–61, 2018.
- [26] “M2mqtt for unity.” (), [Online]. Available: <https://github.com/gpvigano/M2MqttUnity>.
- [27] A. M. Madni, C. C. Madni, and S. D. Lucero, “Leveraging digital twin technology in model-based systems engineering,” *Systems*, vol. 7, no. 1, p. 7, 2019.
- [28] minhcl. “Quaternion và phép quay không gian – phần 1.” (), [Online]. Available: <https://minhcl.wordpress.com/2014/01/30/quaternion-va-phep-quay-khong-gian-phan-1/>.
- [29] ohstem. “Đào tạo huấn luyện điều khiển cánh tay robot bằng công nghệ vr.” (), [Online]. Available: <https://onetech.vn/works/nghien-cuu-phat-trien/dao-tao-huan-luyen-dieu-khien-canhang-tay-robot>.
- [30] “Openxr plugin.” (), [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.xr.openxr@1.11/manual/index.html>.
- [31] M. Ouadoudi, M. H. Schwarz, and J. Borcsok, “Development of a digital twin for an elevator system,” *WSEAS Transactions on Systems and Control*, vol. 17, pp. 214–221, 2022.
- [32] A. Owen-Hill. “Introduction to robodk’s api: How to automate repetitive tasks.” (), [Online]. Available: <https://robodk.com/blog/introduction-to-robodks-api-how-to-automate-repetitive-tasks/>.
- [33] “Progressbar pack.” (), [Online]. Available: <https://assetstore.unity.com/packages/tools/gui/progressbar-pack-120981>.
- [34] ROS Development Team, *Understanding ROS Nodes*, <https://wiki.ros.org/ROS/Tutorials/UnderstandingNodes>, Year of access.

- [35] “Sadp tools.” (), [Online]. Available: <https://www.hikvision.com/en/support/tools/hitools/clea8b3e4ea7da90a9/>.
- [36] J. P. Sáenz, “An introduction to micro electro mechanical systems (mems),” *BURAN*, vol. N°22, Sep. 2005.
- [37] M. Singh, E. Fuenmayor, E. P. Hinchy, Y. Qiao, N. Murray, and D. Devine, “Digital twin: Origin to future,” *Applied System Innovation*, vol. 4, no. 2, p. 36, 2021.
- [38] “Small robot arm.” (), [Online]. Available: <https://github.com/SkyentificGit/SmallRobotArm>.
- [39] space3d. “Ứng dụng của thực tế ảo trong các lĩnh vực ngày nay.” (), [Online]. Available: <https://space3d.vn/page/ung-dung-cua-thuc-te-ao-trongcac-linh-vuc-ngay-nay-131.html>.
- [40] R. N. TUTORIALS. “Mqtt (mq telemetry transport).” (), [Online]. Available: <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>.
- [41] “Ui blur.” (), [Online]. Available: <https://assetstore.unity.com/packages/vfx/shaders/ui-blur-173331>.
- [42] “Unity’s interface.” (), [Online]. Available: <https://docs.unity3d.com/Manual/UsingTheEditor.html>.
- [43] S. Vaidya, P. Ambad, and S. Bhosle, “Industry 4.0 – a glimpse,” *Procedia Manufacturing*, 2018, ISSN: 2351-9789.
- [44] J.-N. Voigt-Antons, T. Kojic, D. Ali, and S. Möller, “Influence of hand tracking as a way of interaction in virtual reality on user experience,” in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020, pp. 1–4. DOI: 10.1109/QoMEX48832.2020.9123085.
- [45] W3. “Ua server rtsp communication.” (), [Online]. Available: https://www.w3.org/2008/WebVideo/Fragments/wiki/UA_Server_RTSP_Communication.
- [46] “What are the azimuth and elevation of a satellite?” (), [Online]. Available: <https://www.celestis.com/resources/faq/what-are-the-azimuth-and-elevation-of-a-satellite/>.
- [47] “Xr interaction toolkit.” (), [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.5/manual/index.html>.

- [48] Y. Zheng, S. Yang, and H. Cheng, “An application framework of digital twin and its case study,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 1141–1153, 2019.

THÔNG TIN SINH VIÊN

Danh sách tác giả Đồ Án:

1. **Nguyễn Phúc Tiên** - ID: 2014725

- Số điện thoại: (+84)944.726.281
- Email: tien.nguyenphuc2612@hcmut.edu.vn

2. **Nguyễn Minh Tiên** - ID: 2014722

- Số điện thoại: (+84)949.291.518
- Email: tien.nguyenminhtien1@hcmut.edu.vn

3. **Nguyễn Văn Thịnh** - ID: 2014603

- Số điện thoại: (+84)974.920.907
- Email: thinh.nguyen2002a1@hcmut.edu.vn