

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI
CORSO DI LAUREA TRIENNALE IN INGEGNERIA MECCANICA E
MECCATRONICA

TESI DI LAUREA TRIENNALE

**INTERAZIONE SICURA TRA UOMO E ROBOT
IN UNO SPAZIO DI LAVORO
COLLABORATIVO**

Relatore: Dr. Stefano Ghidoni

Laureando: Nicolò Alvise Meloncelli
1049716-IMM

ANNO ACCADEMICO: 2016-17

SOMMARIO

La collaborazione tra uomo e robot è un tema d'attualità nella realtà industriale tanto da essere uno degli aspetti fondamentali di Industria 4.0. Secondo le recenti normative, il robot non deve più essere necessariamente separato dall'uomo e isolato in recinti, ma si può valutare uno spazio di lavoro collaborativo in cui uomo e robot coesistono. L'interazione tra uomo e robot è possibile grazie alla presenza sul mercato di sensori sofisticati con elevata performance sulla sicurezza, inoltre i dispositivi per usufruire della realtà aumentata, disponibili in numero sempre crescente, rendono più efficace lo svolgimento dei compiti collaborativi. Durante l'esecuzione delle operazioni collaborative è fondamentale monitorare l'uomo per questioni di sicurezza, di efficienza e per l'apprendimento: il robot, tramite tecniche di Machine Learning, può imparare dall'uomo a eseguire una sequenza di azioni. Per simulare le operazioni collaborative è utile ricavare un modello virtuale dell'uomo, al fine di migliorare sicurezza e flessibilità. In questa tesi si fornisce, anzitutto, una descrizione sintetica del contesto in cui si inserisce l'argomento trattato, tale contesto è Industria 4.0. Sempre nell'introduzione, si riassume il percorso storico, dagli albori della robotica ai tempi odierni, della normativa sulla sicurezza che disciplina l'interazione tra uomo e robot. Successivamente, vengono citati alcuni sensori utili alla collaborazione tra uomo e robot: sensori ottici, occhiali per la realtà aumentata, il braccialetto di controllo per i gesti, un sensore di prossimità, i sensori Empatica E3 e Zephyr BioHarness3, e il dispositivo Microsoft Kinect. Il primo capitolo riguarda l'interazione sicura tra uomo e robot e comprende: le operazioni collaborative descritte dalla normativa, sistemi di localizzazione inerziali, campi di sicurezza dinamici generati da scanner di tipo laser, la coordinazione tra uomo e robot mediante la demarcazione di zone sulla superficie dello spazio di lavoro tramite il dispositivo SafetyEye di Pilz, si accenna a sistemi di localizzazione tramite onde radio e si tratta molto brevemente la ricostruzione della posa dell'operatore tramite telecamere stereo e sensori inerziali. Nel secondo capitolo vengono riportati alcuni schemi di alto livello che guidano a implementare soluzioni collaborative reali e una sintesi della tecnica di apprendimento automatico Learning from Demonstration.

INDICE

1	INTRODUZIONE	1
1.1	Collaborazione tra uomo e robot come obiettivo dell'Industria 4.0	1
1.2	Percorso storico	2
1.3	Sensori per la collaborazione tra uomo e robot	3
2	INTERAZIONE SICURA TRA UOMO E ROBOT	11
2.1	Operazioni collaborative previste dalla normativa	11
2.2	Misure di sicurezza per la collaborazione tra uomo e robot	14
2.3	Sistemi di localizzazione per la collaborazione tra uomo e robot	22
3	SCHEMI PER LA COLLABORAZIONE TRA UOMO E ROBOT	29
3.1	Architetture di alto livello per compiti collaborativi	29
3.2	Tecniche di Apprendimento Automatico	34
	Conclusioni	41
	BIBLIOGRAFIA	43

ELENCO DELLE FIGURE

Figura 1	I nove pilastri dell'Industria 4.0.	2	
Figura 2	Hardware per la collaborazione tra uomo e robot che comprende: un dispositivo di visione che integra scanner laser, telecamera a tempo di volo, un sistema stereo di telecamere con unità pan-tilt; occhiali alla realtà aumentata, biosensori, un braccialetto di controllo per i gesti, un microfono, un tablet, delle casse, il dispositivo Microsoft Kinect e sul braccio del robot sensori di prossimità.	4	
Figura 3	Occhiali per la realtà aumentata.	5	
Figura 4	I biosensori Empatica E3 e Zephyr BioHarness3.		5
Figura 5	Esempio di braccialetto di controllo.	6	
Figura 6	a) Matrice 3×4 di sensori costituiti da condensatori integrati con una linea di transistor elettrochimici connessi a un display elettrocromico. b) Attivazione dei segmenti del display mediante le radiazioni infrarosse del dito. c) Attivazione dei segmenti del display con la luce di un puntatore laser.	6	
Figura 7	Il dispositivo Microsoft Kinect.	7	
Figura 8	Sphere-Swept Line.	15	
Figura 9	Axis aligned bounding box.	15	
Figura 10	Gerarchia dei volumi confinanti.	16	
Figura 11	Processo di generazione del campo attorno al braccio. (a) La posizione delle sfere è valutata secondo la cinematica diretta del manipolatore. (b) L'impronta delle sfere è ottenuta mediante la loro proiezione sul piano bidimensionale. (c) Il campo del braccio è fuso con quello della piattaforma.	19	
Figura 12	Processo di verifica della visibilità. (a) Il campo dopo l'operazione di and logico. Alcuni punti del campo non sono visibili. (b) Visibilità del poligono del primo sensore. (c) Visibilità del poligono del secondo sensore. (d) Il campo finale: l'unione dei due poligoni visibili.	20	
Figura 13	Un operatore è sulla traiettoria della piattaforma. Il warning field rileva la sua presenza e la velocità del robot viene ridotta.	21	

Figura 14	Scansione della <i>warning zone</i> e della <i>detection zone</i> in una fase di lavoro non collaborativa.	21
Figura 15	Creazione del corridoio perché l'operatore esegua l'operazione collaborativa.	22
Figura 16	Il mass-spring system.	23
Figura 17	Da sinistra il mass-spring system, il cylinder model e la proiezione del cylinder model.	24
Figura 18	Relazioni tra i particle filters e il mass-spring system.	24
Figura 19	L'immagine di sinistra mostra la griglia di campionamento per le braccia visualizzate con dei calcoli sulla media. I campioni della superficie mostrano la qualità con cui combaciano griglia e arti dell'uomo con una scala dal rosso al verde. I punti del campione occlusi sono gialli, mentre quelli del contorno sono bianchi o neri. L'immagine di destra mostra varie prove con ambiente e vestiti diversi.	25
Figura 20	Gerarchia dello scheletro del GypsyGyro-18.	26
Figura 21	Il dispositivo Ubisense.	26
Figura 22	Sistemi di riferimento del GypsyGyro-18 e dell'Ubisense.	27
Figura 23	Architettura basata su ROS.	29
Figura 24	Finestra che abilita l'operatore a eseguire e terminare l'operazione collaborativa.	30
Figura 25	Componenti del modulo di riconoscimento.	30
Figura 26	Schema dell'architettura di alto livello JAHIR.	31
Figura 27	Posizionamento delle telecamere nella configurazione sperimentale.	33
Figura 28	Il P5 data glove.	34
Figura 29	Tecniche di derivazione della policy.	36
Figura 30	Processo in cui viene mappata la dimostrazione dall'insegnante al soggetto che la deve apprendere.	37
Figura 31	Demonstration e imitation, identity e non-identity per embodiment mapping e record mapping.	38

ELENCO DELLE TABELLE

Tabella 1	Le librerie OpenNi e Microsoft SDK a confronto.	8
-----------	---	---

INTRODUZIONE

COLLABORAZIONE TRA UOMO E ROBOT COME OBIETTIVO DELL'INDUSTRIA 4.0

La nuova Industria 4.0 si fonda su nove pilastri fondamentali: *simulation, horizontal and vertical system integration, the industrial internet of things, cybersecurity, the cloud, additive manufacturing, augmented reality, big data and analytics, autonomous robots* [1], riportati in figura 1. Per quanto riguarda il primo punto, simulation, i dati sugli operatori, sui mezzi di produzione e sui prodotti vengono impiegati per simulare nuovi settaggi dei macchinari in un mondo virtuale prima del fisico changeover. Riguardo a horizontal and vertical system integration, ci sarà maggior coesione tra compagnie, dipartimenti, funzioni e competenze. Ciò prevede un maggior scambio di informazioni tra compagnie, fornitori e venditori. Il terzo punto, the industrial internet of things, comporta l'inserimento di computer embedded all'interno dei macchinari e un sistema che consente di connetterli a internet in modo da avere un controllo centralizzato dei mezzi in produzione. Con l'aumentare dei mezzi connessi a internet diventa estremamente importante il tema della sicurezza, cybersecurity, per far ciò saranno impiegati, per esempio, sistemi di autenticazione più sofisticati. Tramite la connessione a internet è possibile accedere a servizi web per depositare dati e utilizzare applicazioni che monitorano e controllano i processi, la reattività di tali piattaforme dovrà scendere a pochi millisecondi, questo è il quinto punto: the cloud. Per quanto riguarda additive manufacturing, le stampanti 3D avranno un ruolo fondamentale per produrre componenti e prototipi. Inoltre, l'analisi condotta su un vasto insieme di dati, big data analytics, consentirà di risparmiare energia e ottimizzare la qualità della produzione. La realtà aumentata, augmented reality, permetterà agli operatori di usufruire comodamente di informazioni e di ricevere istruzioni sul posto di lavoro, indossando, per esempio, occhiali che sfruttano questa tecnologia. Il punto fondamentale per questa tesi è autonomous robots, cioè il fatto che i robot stanno diventando più flessibili, autonomi e collaborativi, lavoreranno vicino agli operatori e impareranno da loro, costeranno meno e saranno più abili rispetto ai robot attuali. Uomo e robot coesisteranno in uno spazio di lavoro comune non perché il robot sostituisca l'uomo, ma perché lo assista in modo che ci sia una maggior efficienza ed efficacia nell'eseguire un dato compito. L'operatore dell'Industria 4.0 sfrutterà l'automazione come complemento alle proprie capacità sensoriali, fisiche e cognitive attraverso

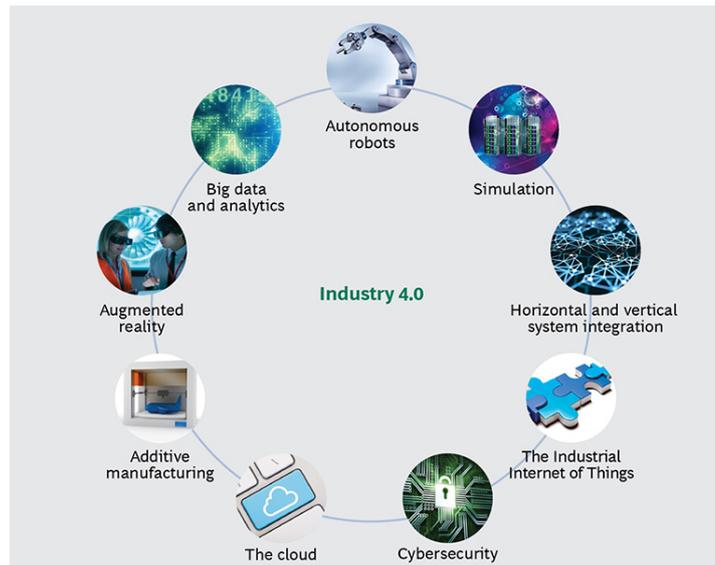


Figura 1: I nove pilastri dell'Industria 4.0.

la *cyber-physical system integration* [2]. Un cyber-physical system è un sistema progettato per migliorare l'interazione dinamica dell'operatore con le macchine attraverso interfacce uomo-macchina intelligenti e migliorando la percezione degli organi di senso e delle capacità cognitive, per esempio, indossando data gloves e occhiali per la realtà aumentata. La collaborazione tra uomo e robot risulta quindi essere un importante obiettivo dell'Industria 4.0.

PERCORSO STORICO

Fin dagli albori della robotica, la sicurezza ha rivestito un ruolo fondamentale [3]. I primi robot industriali azionati idraulicamente erano grandi e potenti, i controlli erano semplici, e non molto affidabili. L'interesse per la sicurezza dei lavoratori portò all'ovvia conclusione di isolare i robot tramite recinti e di non permettere a nessuno di avvicinarsi. I primi robot evitarono all'uomo compiti pericolosi, faticosi e sporchi in fabbrica. Le condizioni lavorative degli operatori migliorarono molto, esempi in fonderia sono i processi di stampatura e forgiatura. Per assicurare la sicurezza sul posto di lavoro, si iniziarono a codificare i requisiti di sicurezza per gli uomini che lavorano attorno ai robot industriali negli Stati Uniti e in Europa. Negli Stati Uniti, la Robotic Industries Association (RIA) sviluppò lo standard di sicurezza per i robot R15.06 tramite l'American National Standard Institute (ANSI). In Europa, l'ISO portò la prima edizione dell'ISO 10218 nel 1992, che fu successivamente adottata dal CEN come EN 775.

I progressi nella robotica continuarono, furono introdotti i robot azionati elettricamente con *servo-control*. Queste macchine andarono

avanti a trasformare molti lavori industriali che richiedevano precisione e ripetibilità, soprattutto nelle saldature, sia a punti che ad arco.

I robot continuarono a essere isolati dagli operatori, tuttavia fu data più attenzione al bisogno degli uomini di interagire con loro, particolarmente negli interventi di manutenzione e settaggio. Poiché ciascuna cella che contiene robot è resa unica dal processo produttivo in cui è impiegata, si comprese che la valutazione del rischio: cioè la valutazione per ciascuna applicazione sulla probabilità e sulla gravità che un certo evento pericoloso possa verificarsi a danno dell'operatore e le misure di sicurezza per evitarlo, era necessaria e fu inserita nell'ANSI/RIA R15.06-1999. La robotica continuò ad evolversi, ma col nuovo millennio sono stati sviluppati così tanti elementi di controllo per i robot che era tempo di cambiare gli standard di sicurezza e provvedere a una nuova e migliore guida per l'interazione dell'uomo con i robot industriali.

Prima fu pubblicata nel 2006 l'ISO 10218-1 dedicata ai soli robot, si continuò nel 2011 sviluppando requisiti di sicurezza per le celle robotiche e l'integrazione, che è l'installazione di accessori per il robot o il fatto di aggiungere al robot un'altra macchina per formare un sistema utile alla produzione, questo lavoro fu pubblicato come ISO 10218-2:2011 assieme alla seconda edizione dell'ISO 10218-1:2011, tali norme sono armonizzate nell'Unione Europea. Negli Stati Uniti e in Canada sono state elaborate l'ANSI/RIA R15.06 e CAN/CSA Z434 rispettivamente. L'arrivo di nuovi software per i controlli *safety-rated*, cioè caratterizzati da una funzione di sicurezza prestabilita con una relativa performance sulla sicurezza [4], dedicati ai robot industriali rende possibili nuove applicazioni e permette l'introduzione di nuove possibilità nel campo dell'automazione: la cooperazione tra uomo e robot. Il robot non è più isolato dietro un recinto, all'uomo è permesso accedere allo spazio di lavoro collaborativo.

SENSORI PER LA COLLABORAZIONE TRA UOMO E ROBOT

Per aumentare la flessibilità dell'ambiente produttivo sono necessarie nuove strategie innovative, una di queste è permettere la collaborazione tra uomo e robot; in tal modo la forza e la precisione del robot fungono da complemento alle capacità cognitive dell'uomo. Rendere la collaborazione efficiente e sicura comporta molte sfide, tra cui: la percezione del robot, la pianificazione dinamica delle operazioni, lo scambio di informazioni tra uomo e robot, la valutazione dello stato dell'operatore e il rispetto degli standard di sicurezza [5]. Riguardo al primo aspetto, ovvero la percezione del robot, si possono usare sensori di coppia, visivi e di prossimità in modo da monitorare lo spazio di lavoro, la posizione dell'operatore, l'energia trasferita durante un contatto tra robot e uomo [6, 7]. Sul secondo aspetto: pianificazio-

ne dinamica delle operazioni, tramite l'osservazione dello sguardo dell'uomo mediante appositi occhiali, per esempio, il sistema di controllo del robot può prevedere la prossima azione dell'operatore e stabilire in anticipo quella del robot [8, 9, 10]. Lo scambio di informazioni tra uomo e robot può avvenire attraverso comandi vocali [11], teleoperazione [12] e tecniche di apprendimento automatico come, per esempio, la tecnica *Learning from Demonstration* [13]. A tal fine saranno utili interfacce acustiche e gestuali [14], dispositivi basati sulla realtà aumentata interconnessi al sistema di controllo per ricevere informazioni sulla presente e successiva operazione e possibili pericoli durante il loro svolgimento, microfoni e bracciali per identificare i gesti dell'uomo. Per monitorare lo stato dell'operatore possono essere usati biosensori e occhiali capaci di analizzare il suo sguardo. Infine gli standard di sicurezza definiscono delle modalità operative che saranno trattate nel prossimo capitolo di questa tesi: esse non rappresentano le uniche possibilità, infatti sono ammesse alternative purché rispettino i requisiti di sicurezza della normativa.

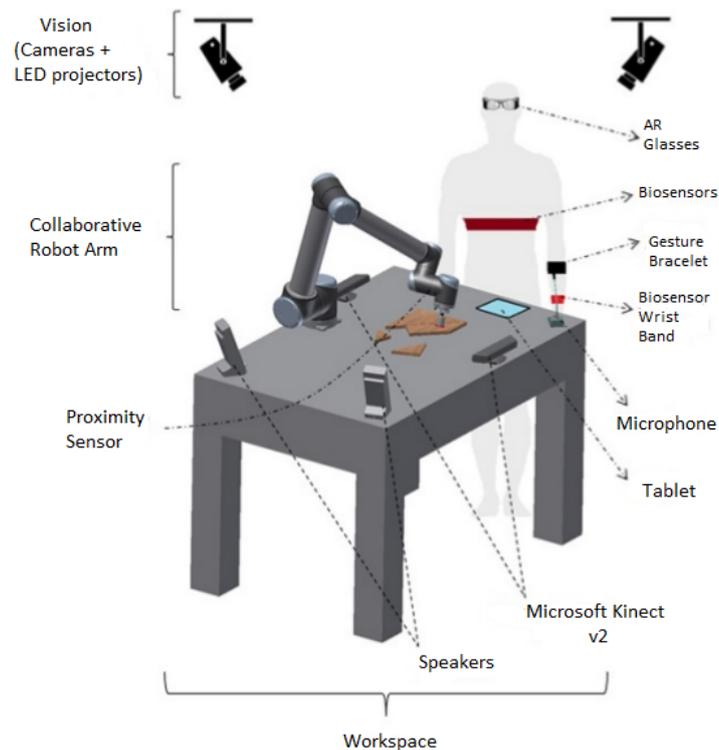


Figura 2: Hardware per la collaborazione tra uomo e robot che comprende: un dispositivo di visione che integra scanner laser, telecamera a tempo di volo, un sistema stereo di telecamere con unità pan-tilt; occhiali alla realtà aumentata, biosensori, un bracciale di controllo per i gesti, un microfono, un tablet, delle casse, il dispositivo Microsoft Kinect e sul braccio del robot sensori di prossimità.

Partendo dall'alto, in figura 2 sono rappresentati in modo generico sensori visivi: l'obiettivo secondo [5] è quello di avere un sistema che integra in un solo dispositivo scanner laser, per mappare su larga scala il luogo di lavoro, una telecamera a tempo di volo, per una prima caratterizzazione della scena, e un sistema stereo che comprende due telecamere, proiettori a led e unità pan-tilt.



Figura 3: Occhiali per la realtà aumentata.

L'operatore indossa occhiali che sfruttano un framework di realtà aumentata (figura 3) per essere informato in tempo reale sul lavoro da svolgere, evitando un eccessivo carico cognitivo e distrazioni; inoltre, utilizzando sia gli occhiali alla realtà aumentata che il dispositivo Microsoft Kinect (figura 7), è possibile stimare dove si focalizza l'attenzione dell'operatore [10], ciò serve a stabilire in anticipo quale sarà la prossima azione del robot [5].



(a) Empatica E3.



(b) Zephyr BioHarness3.

Figura 4: I biosensori Empatica E3 e Zephyr BioHarness3.

I biosensori: sia la fascia che il braccialetto, servono a monitorare i fattori psicofisiologici [15] dell'operatore durante le operazioni collaborative, per avere informazioni sul carico di lavoro e per sapere se l'operatore ha fiducia sulla sicurezza delle modalità operative [5]. Il dispositivo Empatica E3 ha sensori quali: elettrodi per l'attività elettrodermica, un fotopletismografo e un termometro ottico a infrarossi [16], capaci di misurare rispettivamente la conduttanza cutanea, la frequenza cardiaca e la temperatura periferica. Tali parametri psicofi-

siologici sono utili per verificare se l'operatore è in uno stato di ipertensione [17]. Il biosensore Zephyr BioHarness3 è usato in [18] per studiare le condizioni in cui si verificano i colpi di sonno, in modo da prevenirli. Da tale studio emerge che l'operatore prima di assopirsi ha un calo della frequenza respiratoria e un aumento dei battiti cardiaci, tali parametri psicofisiologici sono rilevati mediante un sensore di pressione e degli elettrodi per l'elettrocardiogramma.



Figura 5: Esempio di braccialetto di controllo.

Il braccialetto di controllo, secondo [19], tramite un microcontrollore interagisce con un pc o un dispositivo mobile con la connessione Bluetooth; inoltre il microcontrollore legge gli input dei sensori: accelerometri, giroscopi, GPS o WLAN per misurare la posizione globale, trasmettitore e lettore RFID per rilevare la prossimità a un oggetto, un emettitore a infrarossi e un compasso digitale per verificare a cosa sta puntando l'operatore. Il pc tramite il braccialetto di controllo rileva i movimenti del polso e delle mani e fornisce un certo feedback; con tale dispositivo si vuole rendere la comunicazione tra uomo e robot più immediata [5]. In [20] il braccialetto di controllo Myo (figura 5) è usato per controllare un braccio robotico virtuale, questo braccialetto oltre a accelerometri e giroscopi ha sensori per l'elettromiografia (EMG): questi dispositivi rilevano l'attività elettrica a valle dei muscoli su cui è posto il braccialetto.

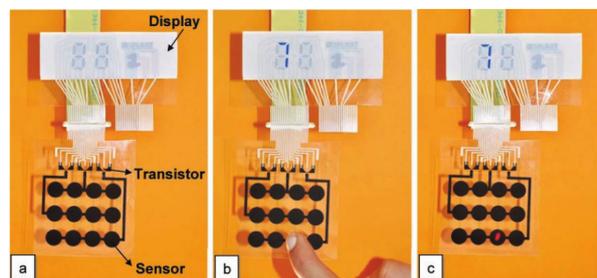


Figura 6: a) Matrice 3×4 di sensori costituiti da condensatori integrati con una linea di transistor elettrochimici connessi a un display elettrocromico. b) Attivazione dei segmenti del display mediante le radiazioni infrarosse del dito. c) Attivazione dei segmenti del display con la luce di un puntatore laser.

Come sensori di prossimità da installare sul braccio meccanico del robot, in [5] sono proposte pellicole multistrato che si ottengono stampando materiali appositamente selezionati [7]. Nel caso del sensore di pressione presentato in [7] (figura 6) ci sono cinque strati di materiale, in particolare il primo è un fluoropolimero e serve come dielettrico per il condensatore, in modo da avere un sensore piezoelettrico, poiché il polimero è sensibile alle variazioni di pressione e di temperatura e si polarizza, comportando una differenza di potenziale ai capi delle armature. Gli altri strati servono a realizzare un transistor, utile alla lettura dei dati del sensore, poiché il transistor ne amplifica il segnale e costituisce un'uscita a bassa impedenza, e un display elettrocromico (ECD) che visualizza l'output del sensore ed è alimentato in corrente dal transistor. Il transistor ha il gate connesso a un elettrodo del sensore in modo che sia uno switch per la corrente che alimenta il display. Il sensore rileva la presenza dell'uomo tramite le radiazioni emesse dal corpo fino a 0,2 m.

Il tablet serve come interfaccia acustica per i comandi vocali, la voce è catturata da un microfono e il dialogo è interpretato dalle applicazioni del tablet e formulato mediante un dialogue manager, usando le casse perché la voce sintetica del dialogue manager sia udibile dall'operatore [5]. Il tablet è usato anche come interfaccia gestuale [5], può essere connesso col braccialetto di controllo.



Figura 7: Il dispositivo Microsoft Kinect.

Il Microsoft Kinect è un sensore RGB-D [21], quindi fornisce immagini a colori e in uno spazio tridimensionale. L'hardware del dispositivo comprende una telecamera RGB con risoluzione base di 640×480 pixel a 30 Hz, oppure, attivando un'opzione, di 1280×1024 pixel a 10 Hz; inoltre vi è un sensore di profondità: un proiettore a infrarossi

e una telecamera a infrarossi che creano una mappa che fornisce la distanza tra un oggetto e la telecamera in un intervallo compreso tra 0,8 m e 3,5 m, il video in uscita ha risoluzione di 640×480 pixel e la frequenza di fotogrammi è di 30 Hz, l'angolo del campo di vista è di 57° in orizzontale e 43° in verticale. Vi è poi un vettore di quattro microfoni per ricevere il segnale audio e un'unità tilt motorizzata per spostare il sensore verticalmente di 27° , mantenendo costante l'asse orizzontale [21].

Tramite i tool software sviluppati, il dispositivo riesce, per esempio, a catturare il movimento dell'uomo nelle tre dimensioni, a riconoscere la voce dell'uomo e a identificarne la faccia [21]. Alcune delle librerie per Microsoft Kinect sono: *OpenNi*¹, che è un tool open-source e multiplatforma, *Microsoft Kinect SDK*², che è stata rilasciata da Microsoft ed è disponibile solo per Windows, e *OpenKinect*³, che è una libreria open-source mantenuta da una comunità aperta di persone [21]. Molte funzioni di queste librerie sono paragonabili

	OpenNi	Microsoft SDK
Camera calibration	✓	✓
Automatic body calibration		✓
Standing skeleton	✓ (15 joints)	✓ (20 joints)
Seated skeleton	X	✓
Body gesture recognition	✓	✓
Hand gesture analysis	✓	✓
Facial tracking	✓	✓
Scene analyzer	✓	✓
3-D scanning	✓	✓
Motor control	✓	✓

Tabella 1: Le librerie OpenNi e Microsoft SDK a confronto.

tra loro (vedi tabella 1), però vi sono delle differenze: per esempio OpenNi richiede che l'utente rimanga in posa per un certo tempo, che dipende dalla potenza del processore e dalle condizioni dell'ambiente, per identificare abbastanza giunti in modo da tracciarne lo scheletro; mentre per Microsoft SDK il tempo di calibrazione è nullo e questo tool riesce a identificare la posa dell'operatore anche quando è seduto. Inoltre, OpenNi si focalizza sul tracciamento delle mani, mentre Microsoft SDK riconosce semplici gesti come spingere o stringere. Tuttavia la versione OpenNi(2.0) permette anche l'installazione

¹ <https://github.com/OpenNI/OpenNI>

² <https://developer.microsoft.com/en-us/windows/kinect>

³ <https://github.com/OpenKinect/libfreenect/>

di Microsoft SDK sulla stessa macchina, usando il driver Microsoft Kinect [21].

L'errore tipico nel tracciare lo scheletro dell'uomo in base alla postura assunta è di 0,1 m [21]. I dati ottenuti dal Microsoft Kinect devono essere preprocessati prima di essere elaborati dagli algoritmi per computer vision, perché è necessario allineare spazialmente l'output della telecamera RGB e quello della telecamera a infrarossi. Inoltre i dati vettoriali sulla profondità sono molto distorti, molti pixel dell'immagine potrebbero non avere profondità a causa delle riflessioni multiple, oggetti trasparenti o della dispersione per capelli e tessuti. Queste lacune sulla profondità devono essere colmate prima di usare i dati, perciò molti sistemi basati su Microsoft Kinect partono con un modulo di preprocessing che attua la ricalibrazione della telecamera e/o il filtraggio dei dati di profondità. Il dispositivo è stato calibrato in fabbrica e i parametri della telecamera sono conservati nella memoria del dispositivo, tuttavia tale calibrazione è adeguata per un uso casuale come il tracciamento di oggetti, ma non accurata per ricostruire una mappa 3-D per la quale deve essere individuata una nuvola di punti nello spazio tridimensionale più precisamente. Inoltre non viene corretta la distorsione sulla profondità, è impossibile quindi recuperare dei dati [21]. In [22, 23, 24] sono proposte delle soluzioni per calibrare la telecamera del Microsoft Kinect.

I sensori sono stati scelti con ridondanza a favore della sicurezza, infatti più sensori hanno la medesima funzione in modo da evitare guasti e malfunzionamenti del sistema [5].

La tesi è strutturata nel seguente modo:

il capitolo 2 è dedicato all'*Interazione sicura tra uomo e robot* ed è suddiviso nei seguenti paragrafi: *Operazioni collaborative previste dalla normativa* (2.1), in cui sono esposte le modalità collaborative descritte dalla normativa [25]; nel paragrafo 2.2, *Misure di sicurezza per la collaborazione tra uomo e robot*, che è un ampliamento del paragrafo 1.3 dell'introduzione, si trattano argomenti quali l'implementazione sicura della collaborazione tra uomo e robot mediante sensori inerziali in 2.2.0.1 e l'utilizzo di campi di sicurezza dinamici, tramite scanner di tipo laser, per massimizzare la produttività di un robot su piattaforma mobile in 2.2.0.2. Nel paragrafo 2.3, *Sistemi di localizzazione per la collaborazione tra uomo e robot*, si accenna alla localizzazione dell'operatore tramite onde radio, in seguito si tratta di modelli semplificati per produrre uno schema fisico dell'operatore mediante un sistema stereo di telecamere e sensori di tipo inerziale.

Il capitolo 3, *Schemi per la collaborazione tra uomo e robot* è suddiviso nei seguenti paragrafi: 3.1, *Architetture di alto livello per compiti collaborativi*, in cui in 3.1.0.1 si tratta di un'architettura di alto livello basata sul middleware ROS, mentre in 3.1.0.2 si considera un'architettura di alto livello basata sul framework MAROCO e su description logic, e in 3.1.0.3 l'architettura di alto livello in esame è denominata JAHIR.

Infine nel paragrafo [3.2](#), *Tecniche di Apprendimento Automatico*, si descrive in estrema sintesi la tecnica di apprendimento Learning from Demonstration.

OPERAZIONI COLLABORATIVE PREVISTE DALLA NORMATIVA

Nella presente sezione si vogliono introdurre le operazioni collaborative tra uomo e robot descritte dalla normativa [25], esse hanno in comune il fatto di essere eseguite in uno spazio di lavoro collaborativo, ovvero una porzione della cella in cui sia l'uomo che il robot hanno accesso per eseguire un compito determinato. Le operazioni collaborative che verranno descritte possono avvenire singolarmente oppure in combinazione. Ogni falla del sistema di sicurezza deve risultare in uno stop protettivo e l'operazione interrotta potrà iniziare nuovamente dopo un reset che l'operatore deve effettuare esternamente allo spazio di lavoro collaborativo. Gli stop protettivi previsti dalla norma [26] sono quelli di *categoria 0* o di *categoria 1*, il primo comporta un'interruzione immediata di potenza agli attuatori e l'obbligo di dissipare l'energia cinetica accumulata dalle masse rotanti in attrito, mentre il secondo prevede un arresto mantenendo alimentati gli attuatori, si presume che essi aiutino a fermare il movimento della macchina, e sospendendo l'alimentazione appena avvenuto l'arresto. Vi è un terzo stop, tuttavia non è considerato di sicurezza, quello di *categoria 2*: si mantiene l'alimentazione agli attuatori durante la fase di arresto e non si sospende successivamente.

Di seguito sono riportate le operazioni collaborative previste dalla norma [25]:

safety-rated monitored stop: mentre l'operatore è nello spazio di lavoro collaborativo(CWS), il robot non deve muoversi, tuttavia non è necessario togliere l'alimentazione al robot, è sufficiente uno stop di *categoria 2*. Quando l'operatore è uscito dal CWS il robot può riprendere la routine di lavoro senza che ci sia il bisogno di azioni aggiuntive.

hand guiding: il movimento del robot è permesso solo quando l'operatore attiva un apposito dispositivo di guida manuale, da quando l'operatore entra nel CWS finché non afferra la guida il robot deve rimanere in uno stop di *categoria 2*, la velocità del robot deve essere limitata a un valore stimato nella valutazione del rischio.

speed and separation monitoring: il contatto tra l'operatore e il robot è evitato controllando la posizione dell'operatore e adattando

la velocità e/o la posizione del robot per mantenere una distanza di sicurezza che va calcolata secondo [25]:

$$S_p(t_o) = S_h + S_r + S_s + C + Z_d + Z_r, \quad (1)$$

in cui $S_p(t_o)$ è la distanza da mantenere tra l'operatore e il robot in un dato istante t_o , S_h è il contributo dovuto al movimento dell'operatore, S_r è il contributo alla distanza di sicurezza dovuto al fatto che il tempo di reazione del sistema di controllo del robot non è immediato, S_s è lo spazio che il robot percorre dopo che il sistema di controllo ha imposto uno stop, C è la distanza che un arto dell'operatore può percorrere prima che sia rilevato dai sensori, Z_d è l'incertezza sulla posizione dell'operatore, Z_r è l'incertezza sulla posizione del robot.

Il contributo alla distanza di sicurezza dovuto al movimento dell'operatore nello spazio di lavoro collaborativo si calcola con la seguente formula:

$$S_h = \int_{t_o}^{t_o+T_r+T_s} v_h(t) dt, \quad (2)$$

in cui T_r è il tempo di reazione del robot, cioè il tempo impiegato dal sistema di controllo per rilevare la posizione dell'operatore e per attivare lo stop protettivo. T_s è il tempo impiegato dal robot per fermarsi dopo aver iniziato lo stop di *categoria 0* o di *categoria 1*. v_h è la velocità dell'operatore nella direzione del robot, se non è possibile rilevarla si considera $v_h = 1.6 \frac{m}{s}$.

Il contributo alla distanza di sicurezza dovuto al fatto che il tempo di reazione del sistema di controllo del robot non è immediato si calcola con la formula:

$$S_r = \int_{t_o}^{t_o+T_r} v_r(t) dt, \quad (3)$$

in cui v_r è la velocità del robot nella direzione dell'operatore, se non è monitorata deve essere pari alla massima velocità raggiungibile dal robot.

Lo spazio che il robot percorre dopo che il sistema di controllo ha imposto uno stop si calcola con la formula:

$$S_s = \int_{t_o+T_r}^{t_o+T_r+T_s} v_s(t) dt, \quad (4)$$

in cui v_s è la velocità del robot dopo l'attivazione dello stop, essa può essere stimata seguendo le linee guida di [4]: se lo stop avviene in *categoria 0* si effettuano le misure nelle condizioni massime di funzionamento previsto (velocità massima, carico massimo, massima estensione ecc.). Per lo stop in *categoria 1* il tempo di stop e la distanza dipendono dalla velocità, dal carico e dall'estensione, le misure devono essere eseguite al 33%, 66% e 100% di tali parametri, ci si può

limitare al caso 100% se si ha la formula che permette di ricavare i valori intermedi.

I dati Z_d e Z_r , che sono rispettivamente l'incertezza sulla posizione dell'operatore e l'incertezza sulla posizione del robot, dipendono dalle caratteristiche dei sensori utilizzati e devono essere forniti dal produttore. C che è la distanza che un arto dell'operatore può percorrere prima che sia rilevato dai sensori, si calcola secondo le linee guida della norma ISO 13855 [27].

power and force limiting: il contatto tra il robot e l'operatore è considerato un evento normale, ma il tipo di contatto è controllato da misure relative al design del robot e/o di controllo *safety-rated*. In ciascun caso l'obiettivo è quello di limitare le forze che possono impattare il corpo dell'operatore. La norma [25] distingue i contatti tra uomo e robot in *transient contact*, in cui il corpo dell'operatore riesce a ritrarsi dal robot, e in *quasi-static contact*, in cui una parte del corpo dell'operatore rimane bloccata tra il robot e una parte mobile o fissa della cella. Per evitare che il contatto sia dannoso per l'operatore si considerano i limiti relativi a forza e pressione sia per i *quasi-static contact* che per i *transient contact* del *body model*, cioè la rappresentazione del corpo umano tramite segmenti che hanno proprietà biomeccaniche, in ciascuna delle 29 aree distribuite su 12 regioni in cui esso è suddiviso. Secondo la medesima norma [25] l'urto tra uomo e robot si considera completamente anelastico, la massa m_H di ciascuna regione del *body model* è riportata in una tabella, quella del robot m_R va calcolata secondo la formula:

$$m_R = \frac{M}{2} + m_L, \quad (5)$$

in cui m_L è il carico effettivo che include il pezzo lavorato e gli utensili e M è la massa totale della parte mobile del robot. Nella stessa tabella di m_H è fornita la costante elastica k , l'energia trasferita durante l'urto va calcolata con la seguente formula:

$$E = \frac{F_{\max}^2}{2k} = \frac{A^2 p_{\max}^2}{2k}, \quad (6)$$

A è l'area di contatto tra il corpo dell'operatore e il robot, nei casi pratici va considerata l'area di contatto minore, F_{\max} è fornita in tabella per ognuna delle 29 aree delle 12 regioni e lo stesso vale per la massima pressione di contatto p_{\max} . L'energia stimata deve essere minore di un valore anch'esso indicato in una tabella per ciascuna delle 12 regioni, infine la velocità relativa tra il robot e l'operatore v_{rel} può essere calcolata invertendo la formula:

$$E = \frac{F^2}{2k} = \frac{1}{2} \mu v_{rel}^2, \quad (7)$$

in cui μ è la massa ridotta dei due corpi e si calcola con la formula

$$\mu = \left(\frac{1}{m_H} + \frac{1}{m_R} \right)^{-1}, \quad (8)$$

v_{rel} quindi vale:

$$v_{rel} = \frac{F}{\sqrt{\mu k}} = \frac{pA}{\sqrt{\mu k}}. \quad (9)$$

MISURE DI SICUREZZA PER LA COLLABORAZIONE TRA UOMO E ROBOT

Nella presente sezione vengono presentati diversi metodi per l'implementazione della collaborazione sicura tra uomo e robot in diversi scenari produttivi.

Collaborazione sicura tra uomo e robot tramite sistemi di localizzazione inerziali

Secondo [28] metodi di protezione passiva quali giunti elastici e coperture visco-elastiche per il robot sono da evitare perché creano risposte più lente e aumentano le oscillazioni, riducendo l'accuratezza della posizione e della velocità. Sono quindi preferibili sistemi di protezione attiva, come sistemi di controllo che diminuiscono la forza di contatto quando è rilevata una collisione con l'operatore dai sensori di forza/coppia installati sull'*end-effector*, che è il dispositivo del robot progettato perché vi sia attaccata un'interfaccia meccanica. Oppure sistemi di controllo che permettono di evitare le collisioni pianificando una nuova traiettoria per il robot se la distanza dall'operatore è troppo piccola, quest'ultima strategia porta a più flessibilità. Riguardo a quest'ultima possibilità in [28] è descritto un sistema con due componenti principali: il primo monitora l'operatore e il secondo calcola la distanza dal robot. Per realizzare il primo modulo, il movimento dell'operatore è rilevato da sensori inerziali, in quanto sono comodi per le loro piccole dimensioni, robusti ai disturbi elettromagnetici e non sono sensibili all'occlusione ottica. In particolare, per tracciare la posizione degli arti dell'operatore come sensore inerziale viene proposto il *GypsyGyro-18* di *Animazoo*, esso ha un tessuto in *Lycra* sul quale sono attaccati 18 *IMUs* (*Inertial Measurement Units*). Ogni *IMU* è posizionato su un segmento del corpo dell'operatore e provvede a fornire l'orientamento del segmento, mettendo assieme i dati di tutte le misure dei sensori si ottiene un modello dello scheletro dell'operatore. La posizione globale dello scheletro è fornita da un algoritmo iterativo che accumula piccoli errori nel tempo dovuti all'integrazione delle misure inerziali. Per correggere questi errori si aggiunge un sistema di localizzazione globale, è proposto il sistema *Ubisense* che è basato su segnali *UWB* (*Ultra-WideBand*). Da un lato i sensori inerziali fanno grossi errori, maggiori di 0.6 m, per calcolare la posizione globale dell'operatore; il sistema di localizzazione *UWB* ottiene misure più precise con errori minori di 0.15 m. Dall'altro lato la frequenza di campionamento del sistema *UWB* ha un valore massimo di 10 Hz e

non è grande a sufficienza per catturare i movimenti più veloci dell'operatore, mentre i sensori inerziali hanno valori più alti: da 30 Hz a 120 Hz. Questa complementarietà permette di raggiungere migliori risultati: le misure di posizione del sistema UWB correggono l'errore accumulato dal sistema inerziale, mentre quelle di posizione del sistema inerziale riducono la latenza del sistema UWB, fornendo nuovi dati tra una misura e la successiva.

Per quanto riguarda il secondo modulo ciascun arto dell'operatore e membro del robot è coperto con una *SSL* (*Sphere-Swept Line*). In

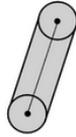


Figura 8: Sphere-Swept Line.

totale sono impiegate diciotto *SSLs* per l'operatore: due per i piedi, per le gambe, per le cosce, per le spalle, per le braccia, per gli avambracci e per le mani, una per l'addome, per il torace, per il collo e per la testa; otto *SSLs* per il robot: una per ogni membro del robot, includendo la base e l'end-effector. Sarebbero necessari 144 test per calcolare la distanza tra il robot e l'operatore, per ridurre il carico computazionale si opta per una gerarchia su tre livelli: il terzo livello comprende tutte le diciotto *SSLs* per l'operatore e le otto per il robot, si usa quando robot e operatore sono molto vicini. Il secondo livello si ottiene coprendo alcune *SSLs* con un *AABB* (*Axis-Aligned Bounding Box*), nel caso dell'operatore sono usati cinque *AABBs*: due sia per gli arti inferiori che per quelli superiori e uno per il torso; nel caso del robot tre: uno per la base e per il primo membro, uno per i successivi due membri e un ultimo per i quattro rimanenti. Il primo livello è ottenuto coprendo l'intero corpo dell'operatore con un *AABB* e facendo lo stesso con la struttura del robot.

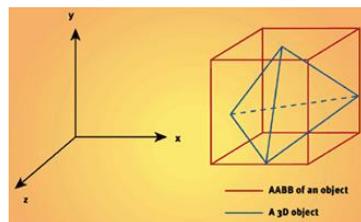


Figura 9: Axis aligned bounding box.

L'algoritmo che calcola la distanza tra robot e operatore considera inizialmente il primo livello della gerarchia, se la distanza è maggiore di una certa soglia non serve considerare gli altri livelli perché robot e operatore sono abbastanza distanti, se invece tale distanza è minore bisogna considerare il secondo livello: il sistema calcola la distanza tra ciascun *AABB* dell'operatore e del robot e stabilisce qual è la distanza

minima tra i due, se la distanza è maggiore di una seconda soglia sono effettuati solo sedici test, uno per il primo livello e quindici per il secondo. Infine se la distanza calcolata dal secondo livello è inferiore rispetto alla soglia prestabilita si effettuano tutti i 144 test, in figura 10 è illustrata la gerarchia sui tre livelli dei volumi confinanti.

Level 1	Level 2	Level 3
Global AABB	Left Lower Limb AABB	Left Thigh SSL Left Leg SSL Left Foot SSL
	Right Lower Limb AABB	Right Thigh SSL Right Leg SSL Right Foot SSL
	Torso-Head AABB	Abdomen SSL Thorax SSL Neck SSL Head SSL
	Left Upper Limb AABB	Left Shoulder SSL Left Arm SSL Left Forearm SSL Left Hand SSL
	Right Upper Limb AABB	Right Shoulder SSL Right Arm SSL Right Forearm SSL Right Hand SSL

(a) Gerarchia dei volumi confinanti per l'uomo.

Level 1	Level 2	Level 3
Global AABB	Base AABB	Base SSL S1 SSL
	Arm AABB	S2 SSL S3 SSL
	Forearm AABB	E1 SSL E2 SSL W1 SSL W2 SSL

(b) Gerarchia dei volumi confinanti per il robot.

Figura 10: Gerarchia dei volumi confinanti.

La più piccola distanza calcolata è usata dal sistema di controllo del robot per attivare le strategie di sicurezza: mantenere il robot in uno stop di *categoria 2* finché la distanza non supera la soglia di sicurezza, o allontanare il robot dall'operatore affinché sia mantenuta la distanza di sicurezza.

In sintesi, ad ogni iterazione il sistema di controllo acquisisce posizione e orientamento degli scheletri dell'uomo e del robot dal sistema di tracciamento. Successivamente questi scheletri sono coperti dai corrispondenti *bounding volumes* (SSLs e AABBs) e viene calcolata la minima distanza tra di essi, questa distanza è comparata con la soglia di sicurezza: se risulta più grande il robot continua la sua operazione, se è più piccola si attiva una strategia di sicurezza.

Massimizzare la produttività di un robot su piattaforma mobile mediante campi di sicurezza dinamici

Usando scanner di tipo laser è possibile definire un numero limitato di zone attorno a un robot che, se violate, comportano da parte del sistema di controllo l'attivazione di una strategia di sicurezza. Queste zone sono note come *safety fields* [29], con la tecnologia attuale se ne possono creare circa 32. Esse vengono configurate off-line, quando l'impianto viene messo in servizio. Gli scanner ammettono due tipi di campi di sicurezza: i *protection field* e i *warning field*. Se un oggetto è rilevato all'interno del protection field il robot è soggetto a uno stop protettivo, se invece è rilevato all'interno del warning field, si adotta una misura di sicurezza meno severa, per esempio viene ridotta la velocità del robot. Scegliendo un campo di sicurezza statico, attraversare una superficie stretta, come quella di un corridoio, oppure avvicinarsi a degli scaffali non è fattibile per un robot su piattaforma mobile, per risolvere questo problema si potrebbe fare uno switch tra un predefinito set di campi. Lo switch può essere avviato dalla velocità o da un segnale esterno, ma questo approccio non garantisce abbastanza flessibilità per coprire tutte le situazioni in cui il robot può trovarsi, a causa del numero limitato di campi di sicurezza configurabili. Un'altra soluzione si basa sul fatto che quando un operatore è rilevato dal warning field il robot può procedere a velocità ridotta, mentre tale problema non si presenta quando la presenza dell'operatore non è rilevata. Considerando questo fatto è possibile optare per campi di sicurezza dinamici, usando *proprioceptive data* e *exteroceptive data* [29]. I proprioceptive data sono la velocità e la posizione del manipolatore, mentre gli exteroceptive data sono forniti dall'ambiente in cui il robot si muove, per esempio la presenza di ostacoli e la distanza da questi.

In [29] si illustra come utilizzare campi di sicurezza dinamici per garantire la sicurezza di un operatore che condivide lo spazio di lavoro con un robot su piattaforma mobile, tutte le formule che seguono sono state tratte da questo articolo. Il profilo del campo di sicurezza viene modificato in base alla velocità della piattaforma, alla posizione del braccio del robot e a fattori esterni, quali la mappa dello spazio di lavoro e la localizzazione del robot in essa. I primi due fattori sono proprioceptive data, mentre l'ultimo è un exteroceptive data.

Maggiore è la velocità della piattaforma, tanto più grande sarà lo spazio necessario per fermarla. La superficie del campo deve almeno coprire lo spazio d_{reaction} dato dalla latenza t_r del sistema di controllo nel rilevare il comando di frenare la piattaforma e lo spazio necessario alla piattaforma per fermarsi $d_{\text{activeBrake}}$, sarà quindi:

$$d_{\text{brake}} = d_{\text{reaction}} + d_{\text{activeBrake}} \quad (10)$$

La distanza percorsa durante t_r è calcolata con la seguente formula:

$$d_{\text{reaction}} = vt_r, \quad (11)$$

in cui v è la velocità del robot e t_r è costante e fornita dal produttore del robot. La distanza $d_{\text{activeBrake}}$ si calcola nel seguente modo:

$$d_{\text{activeBrake}} = vt_b - \frac{1}{2}a_b t_b^2, \quad (12)$$

in cui t_b è il tempo necessario alla piattaforma per fermarsi, esso si ottiene con l'equazione:

$$t_b = \frac{v}{a_b}, \quad (13)$$

la decelerazione a_b imposta dai freni è un dato fornito dal produttore del robot. Dopo aver valutato d_{brake} la dimensione del campo di sicurezza in una generica direzione d è valutata con la formula:

$$s_{dp}(t) = s_{dpmin} + \frac{1}{2} \frac{v_d^2(t)}{a_{db}} + v_d(t)t_r, \quad (14)$$

dove si assume che s_{dpmin} sia la dimensione minima del campo nella direzione d .

Per il calcolo del warning field viene proposta una formula euristica basata sull'interpolazione lineare della velocità del robot tra il suo valore minimo e massimo:

$$s_{dw}(t) = (s_{dwMax} - s_{dwmin}) \frac{v_d(t)}{v_{dMax}} + s_{dwmin}, \quad (15)$$

dove s_{dwMax} e s_{dwmin} sono rispettivamente il massimo e il minimo valore del warning field nella direzione d .

Anche la proiezione del braccio del robot sul piano bidimensionale deve rientrare nel campo di sicurezza per tutelare l'operatore. Per far ciò si copre ciascun asse del robot con una sfera di raggio r , il centro di ciascuna sfera \mathbf{p}_i , con $i = 0, \dots, n$ in cui n è il numero dei link del robot, è posizionato al centro di ciascun asse. Successivamente si considera la posizione di \mathbf{p}_i rispetto alla base b del robot, ottenendo \mathbf{p}_i^b , mediante la seguente trasformazione:

$$\mathbf{p}_i^b = \begin{bmatrix} p_{xi} \\ p_{yi} \\ p_{zi} \\ 1 \end{bmatrix} = \mathbf{T}_i^b \mathbf{p}_i^i, \quad (16)$$

in cui \mathbf{p}_i^i è la posizione del centro della sfera riferita al giunto i in coordinate omogenee, \mathbf{T}_i^b è la matrice di trasformazione omogenea che descrive la posizione e l'orientamento del link i rispetto alla base b . Infine \mathbf{p}_i^b viene proiettato nel piano f tramite le coordinate $\{x, y\}$, ottenendo:

$$\mathbf{p}_i^f = \begin{bmatrix} p_{xi} \\ p_{yi} \end{bmatrix}. \quad (17)$$

Successivamente le impronte delle sfere sul piano bidimensionale vengono fuse in modo da ottenere una regione che circondi la struttura del manipolatore, tale regione viene unita al campo di sicurezza.

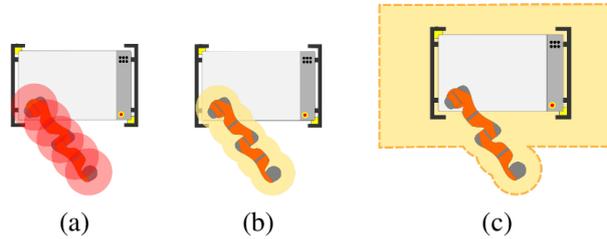


Figura 11: Processo di generazione del campo attorno al braccio. (a) La posizione delle sfere è valutata secondo la cinematica diretta del manipolatore. (b) L'impronta delle sfere è ottenuta mediante la loro proiezione sul piano bidimensionale. (c) Il campo del braccio è fuso con quello della piattaforma.

Oltre a considerare i parametri del robot per la generazione del campo di sicurezza vanno presi in considerazione anche quelli esterni, come gli ostacoli. Essi sono resi noti da una mappa usata dal robot per muoversi nello spazio di lavoro, tale mappa è rappresentata mediante un'immagine che viene processata e, in base ai dati raccolti, si modifica il campo di sicurezza. Gli oggetti noti che violano il campo di sicurezza non rappresentano un pericolo, quindi il robot se è in loro prossimità non è soggetto a uno stop protettivo. La mappa in esame è in gray-scale ed è un file in cui ciascun pixel è codificato in otto bit che deve essere pre-processato per avere un input più semplice da fornire all'algoritmo e per rimuovere il rumore. Il primo passo è quello di convertire la mappa in un file binario attraverso un algoritmo di *thresholding*, successivamente la mappa deve essere processata per eliminare il rumore e definire meglio i contorni utilizzando operazioni morfologiche, come erosione e dilatazione [30]. Dopo aver applicato questi filtri, la mappa è pronta per essere data in input all'algoritmo che ridimensiona il campo: si impostano a uno tutti i punti del campo e si esegue un'operazione di and logico tra i punti del campo e quelli della mappa per eliminare gli ostacoli dal campo. Il contorno della mappa ottenuta dopo questi passaggi è dato in input al *Bresenham's line algorithm* [31] per provare la visibilità del campo. In un campo di sicurezza per sensori 2D, ciascun punto all'interno del campo deve essere direttamente visibile ai sensori. Ciò significa che una linea che connette un punto arbitrario del campo al sensore deve essere totalmente contenuta nel campo. Quando sono disponibili più sensori un punto è direttamente visibile se esso è direttamente visibile da almeno un sensore. L'algoritmo, quindi, partendo dal contorno del campo di sicurezza ipotizzato valuta per ogni sensore il poligono rappresentato dal perimetro del campo visibile e fonde tutti questi poligoni,

formando il campo di sicurezza.

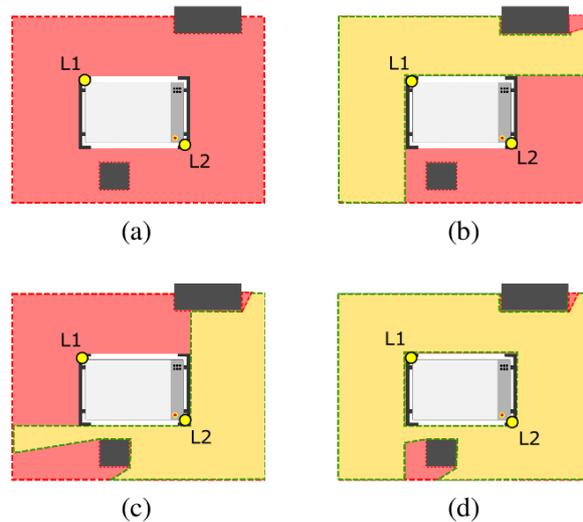


Figura 12: Processo di verifica della visibilità. (a) Il campo dopo l'operazione di and logico. Alcuni punti del campo non sono visibili. (b) Visibilità del poligono del primo sensore. (c) Visibilità del poligono del secondo sensore. (d) Il campo finale: l'unione dei due poligoni visibili.

È opportuno lasciare sempre attivo un campo di sicurezza minimo per evitare che il robot collida con le pareti del luogo di lavoro. Per il braccio meccanico non è previsto un campo di sicurezza minimo, infatti se lo fosse non potrebbe prelevare semilavorati da un banco. Tuttavia deve essere noto l'ostacolo sul quale il robot eseguirà un'operazione. Uno strumento utile per implementare questa soluzione è noto come *semantic map*: la semantic map non è un file binario, contiene informazioni su ogni oggetto registrato come, per esempio, l'altezza. Ciò permette al robot di distinguere un muro da un banco da lavoro.

La prova sperimentale in ambiente industriale con cui è stato testato questo approccio è descritta in [29]. Essa è stata eseguita con il *KUKA omniRob*, rappresentato in figura 13 accanto ad un operatore, il robot è equipaggiato con due scanner laser sui due angoli opposti della piattaforma mobile come in figura 12. Sono stati definiti un warning field e un protection field, quando il robot viola il primo campo la sua velocità viene ridotta del 30%, invece se viene violato il secondo campo si ricorre a uno stop protettivo del robot. Il warning field è dimensionato tenendo conto della velocità del robot, della posizione del braccio meccanico e dei fattori esterni rappresentati nella mappa dello spazio di lavoro, mentre il protection field tiene conto solo dei primi due fattori. Il luogo di test è composto da un corridoio, un locale dedicato alla produzione e un magazzino, il robot deve prelevare il prodotto dall'area di produzione e portarlo nel magazzino.

Con un approccio classico vengono definiti tre campi di sicurezza che si alternano in base alla velocità del robot: al 25%, 50% e 75% della massima velocità, essi sono dimensionati rispettivamente in modo da garantire una distanza di sicurezza per fermare la piattaforma in base al: 50%, 75%, 100% della velocità massima. Il tempo necessario ad attraversare il corridoio con tale configurazione è di 19 s, usando i campi di sicurezza dinamici si riduce a 12 s, ciò comporta una riduzione del 37% del tempo di esecuzione.

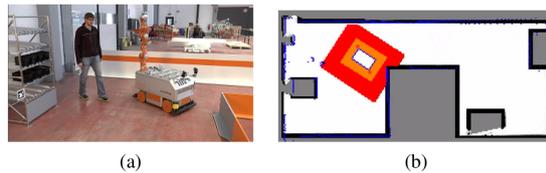


Figura 13: Un operatore è sulla traiettoria della piattaforma. Il warning field rileva la sua presenza e la velocità del robot viene ridotta.

Coordinazione tra uomo e robot mediante la demarcazione di zone sulla superficie dello spazio di lavoro

Una cella di lavoro senza recinti, finalizzata a compiti collaborativi tra uomo e robot, può essere implementata anche usando solo un dispositivo certificato con una certa performance sulla sicurezza: il *SafetyEye* di PILZ [32]. Esso è installato sul soffitto della cella ed è sostanzialmente una telecamera 3D che sorveglia l'area circostante collegata al sistema di controllo del robot. La superficie del pavimento è marcata in giallo nella *Warning Zone*, per avvisare l'operatore che si sta avvicinando al robot, e in rosso nella *Detection Zone*, che se violata comporta uno stop protettivo del robot, come in figura 14, mediante il software fornito da PILZ.

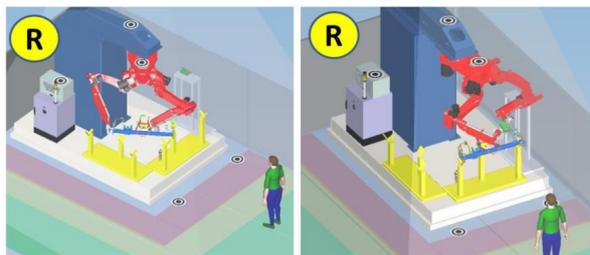


Figura 14: Scansione della *warning zone* e della *detection zone* in una fase di lavoro non collaborativa.

Se, usando lo schema basato su ROS descritto a pagina 29, l'operatore preme il tasto start rappresentato nella figura 24, viene segnalata al sistema di controllo l'intenzione dell'operatore di iniziare l'opera-

zione collaborativa. Per permettere che ciò avvenga si libera un corridoio che permette all'operatore di attraversare la *detection zone* e di raggiungere il robot, come in figura 15.

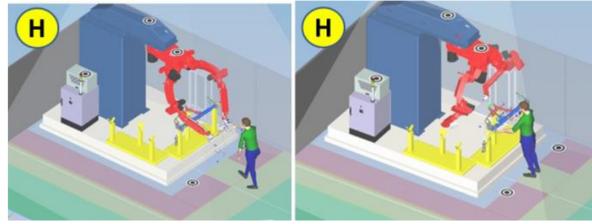


Figura 15: Creazione del corridoio perché l'operatore esegua l'operazione collaborativa.

SISTEMI DI LOCALIZZAZIONE PER LA COLLABORAZIONE TRA UOMO E ROBOT

Un primo contributo alla protezione degli operatori si ottiene monitorando lo spazio di lavoro, in particolare le traiettorie, il movimento degli operatori e l'intenzione di eseguire un certo movimento in modo da poterlo anticipare. Grazie alla disponibilità di nuove classi di dispositivi con performance sulla sicurezza nel campo dell'automazione, come PLC e bus dedicati alla sicurezza, applicazioni in cui è importante salvaguardare l'operatore, come la modalità *speed and separation monitoring*, sono maggiormente supportate. Un metodo che non obbliga l'operatore a indossare radio trasmettitori, qualora si volesse implementare una tecnologia di localizzazione wireless, è quello di analizzare la fluttuazione delle onde elettromagnetiche *radio-frequency* (RF), originate da fitte reti wireless [33]. In tal modo è possibile rilevare la presenza di un operatore e tracciarne la posizione. L'implementazione di questa tecnologia avviene tramite trasmettitori radio embedded che formano una rete *sensor-less* e *device-free localization* (DFL). I vantaggi di tale tecnologia sono economici, poiché tali dispositivi hanno un basso costo e permettono maggiore flessibilità [34]. La tecnologia DFL opera nelle bande libere (2.4 GHz e 5.8 GHz), e diversamente da luce e infrarossi le RF attraversano pareti non metalliche e possono operare in aree con scarsa visibilità [33]. Si presume che la tecnologia DFL sarà presto applicabile tramite gli standard wireless esistenti: WiFi, ZigBee, WirelessHART, ISASP100 [35].

Trovare e tracciare la postura dell'operatore è importante anche per insegnare al robot nuove azioni attraverso dimostrazioni in cui il robot apprende osservando i movimenti. Non è sempre possibile, soprattutto per ragioni economiche, richiedere all'operatore di indossare particolari sensori per identificarne la posa, inoltre il robot di solito ha una visione limitata da una telecamera mono o stereosco-

pica. È possibile ottenere dati sul movimento dell'operatore con un modello 3D e valutando una singola prospettiva dell'operatore [36]. Per implementare tale metodo è consigliata una telecamera stereoscopica e una frequenza di fotogrammi di 10 Hz su un normale pc. Lo scheletro dell'operatore può essere rappresentato, per esempio, da un modello flessibile, il *mass-spring system* [36] riportato in figura 16, che comprende masse puntiformi e aste che rappresentano in modo astratto le ossa: 16 masse e 24 aste, per un totale di $16 \cdot 3 = 48$ gradi di libertà. Il sistema non rappresenta esplicitamente rotazioni: alcune di esse possono essere dedotte dalla posizione delle masse puntiformi. Si assume che le aste non siano completamente rigide, ma flessibili. Tale schema è vantaggioso, poiché il movimento in un certo punto è

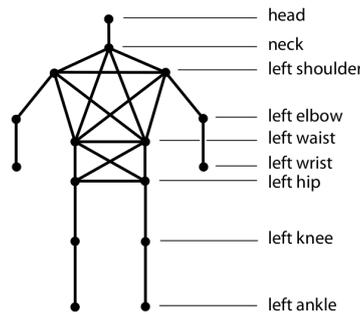


Figura 16: Il mass-spring system.

rappresentato dallo spostamento della rispettiva massa e non richiede di ricorrere alla cinematica inversa per stimarlo. Il modello è migliorato inserendo ulteriori vincoli per limitare lo spostamento angolare degli arti. Poiché le rotazioni non sono definite, le coordinate della struttura sono fornite dall'insieme delle tre posizioni delle masse. Il mass-spring system viene risolto iterativamente, applicando il *Verlet framework* [36]:

$$x_{t+1} = 2x_t - x_{t-1} + a_t \Delta t^2, \quad (18)$$

in cui x_t è la posizione di un punto al tempo t , a_t è l'accelerazione calcolata dalle forze accumulate durante un fotogramma e Δt è l'intervallo di tempo. La velocità viene calcolata implicitamente dalla differenza delle due posizioni precedenti. I pesi delle masse sono scelti in modo che i punti del torso siano più pesanti e quelli delle estremità siano più leggeri, alla testa viene assegnato un peso infinito in modo da renderla immobile: è mossa solo assegnando la posizione rilevata dall'*head tracker*. La resistenza delle aste è stata scelta perché il movimento delle masse sia realistico e lo stato del mass-spring system è usato per calcolare quello del *cylinder model*, in figura 17, che è formato da due cilindri per ogni estremità, due per il torso e uno per la testa. La posizione della testa è tracciata da un *particle filter* che si basa sulla segmentazione del colore della pelle, usando la telecamera stereoscopica. Per il resto del corpo in [36] viene usata una

sola immagine, non è consigliato l'uso della telecamera stereoscopica. Quando il punto della testa nel mass-spring system viene mosso, anche gli altri punti lo sono dell'80% rispetto alla traslazione della testa. Se il modello viene mosso con la testa, la posa del resto del corpo viene ricostruita tracciando le estremità mediante i particle filters e risolvendo il mass-spring system, ottenendo una soluzione implicita per il torso.

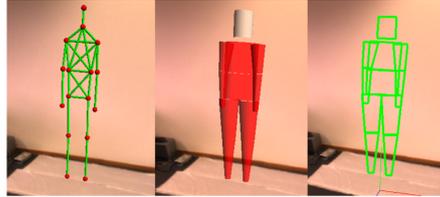


Figura 17: Da sinistra il mass-spring system, il cylinder model e la proiezione del cylinder model.

Il mass-spring system influenza i quattro particle filters delle estremità rappresentati in figura 18. Esso fornisce lo stato dei tre punti riferiti a ciascun arto: per esempio la posizione delle masse puntiformi in corrispondenza della spalla, del gomito e del polso per quanto riguarda un braccio, come input ai particle filters. Questi restituiscono una predizione sullo stato futuro dei punti: tale predizione viene data nuovamente in ingresso al mass-spring system, per correggere eventuali posizioni errate [36]. I quattro particle-filters vengono eseguiti indipendentemente, la massa dei punti che non sono di competenza del particle filter attivo è settata a infinito in modo che tali punti restino immobili. Le stime dei particle filters sono inserite nel mass-spring

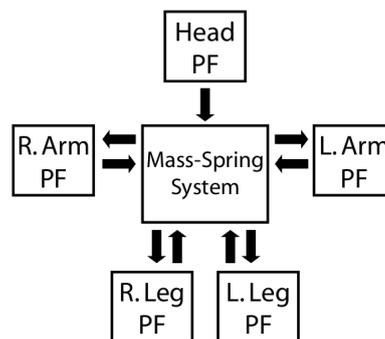


Figura 18: Relazioni tra i particle filters e il mass-spring system.

system contemporaneamente.

Il *measurement model* si ottiene con la proiezione dei cilindri nelle immagini per calcolare due punteggi per particle: un *edge score* e un *surface score* [36]. Il primo si calcola comparando i lati del cilindro con i bordi della superficie dell'immagine e viene assegnato un punteggio maggiore ai bordi che si muovono perché si attribuiscono alla

silhouette dell'operatore. Il surface score si calcola campionando la superficie della proiezione dei cilindri con una griglia regolare come in figura 19. Il punteggio viene attribuito in base al *color model* [36],

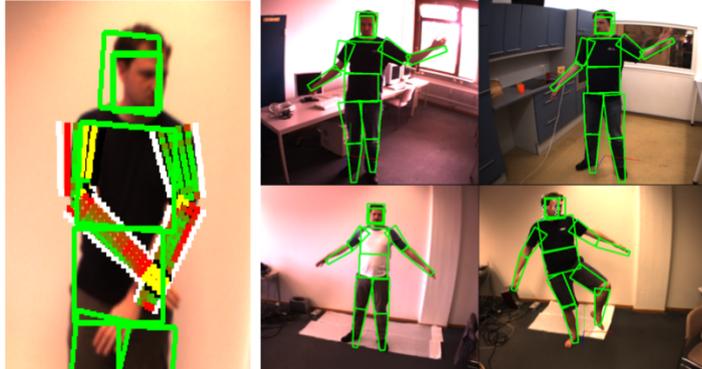


Figura 19: L'immagine di sinistra mostra la griglia di campionamento per le braccia visualizzate con dei calcoli sulla media. I campioni della superficie mostrano la qualità con cui combaciano griglia e arti dell'uomo con una scala dal rosso al verde. I punti del campione occlusi sono gialli, mentre quelli del contorno sono bianchi o neri. L'immagine di destra mostra varie prove con ambiente e vestiti diversi.

questo modello consiste di dieci colori che si riferiscono a un arto rappresentato da un cilindro del *cylinder model*, si assume che il colore rimanga costante solo su un anello attorno al cilindro. Il *color model* per ciascuna parte del corpo viene appreso dalla prima coppia di fotogrammi [36], il colore di un fotogramma viene campionato su una griglia regolare, per i punti che si pensano appartenere all'uomo si effettua la media tra i colori dei vari anelli del cilindro ottenuti da più fotogrammi. Successivamente i colori dei vari anelli vengono interpolati linearmente per avere un colore di riferimento dell'arto, usando come indice la distanza normalizzata dalla base del cilindro.

Il *measurement model* fa uso di un *occlusion model* [36] basato sullo stato attuale del *mass-spring system*, è composto dai quadrilateri formati dalla proiezione dei cilindri. I quadrilateri hanno una profondità minima e massima valutata in una griglia spaziale per verificare se un punto del campione è visibile o meno. Tutti i campioni sono verificati, per l'*edge score* e il *surface score*, si valuta se il campione è occluso o fuori dall'immagine.

In [37] è studiato un sistema di tracciamento tramite sensori inerziali, viene proposto il sistema *GypsyGyro-18* di Animazoo che comprende accelerometri e giroscopi da posizionare sull'operatore. Le posizioni degli arti sono calcolate integrando due volte le accelerazioni fornite dagli accelerometri e gli orientamenti sono ottenuti integrando le velocità angolari ottenute dai giroscopi. Tale sistema è usato per capire come sono orientati gli arti nello spazio perché fornisce posi-

zioni angolari con risoluzione minore di 1° [37], quindi molto precise, per le traslazioni secondo le coordinate (x, y, z) esso accumula errori anche maggiori del 30%, allora è proposto il sistema UWB di Ubisense con risoluzione di 0,3 m. Il GypsyGyro-18 è composto da 18 IMUs (inertial measurement units) poste su un tessuto in Lycra, perché sia indossabile dall'operatore, ciascun IMU è un *InertiaCube3* di *Intersense* che misura le rotazioni attorno agli assi x, y e z dell'arto a cui è attaccato e trasmette tali dati al pc tramite un collegamento wireless. I dati misurati sono riportati su uno scheletro in 3D (figura 20), di tale scheletro il nodo del bacino (hips node) è il più importante e rappresenta la rotazione e la traslazione globale dell'intero corpo.

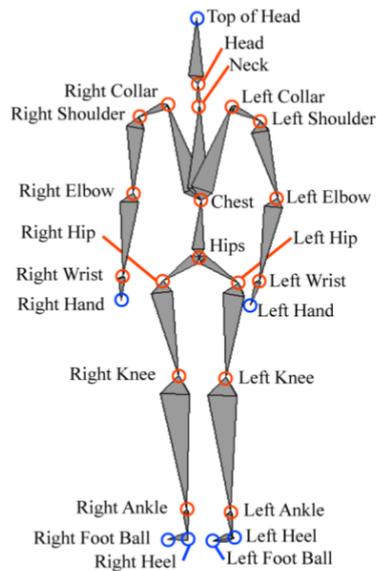


Figura 20: Gerarchia dello scheletro del GypsyGyro-18.

Il dispositivo a UWB (Ultra-WideBand) di Ubisense usato per ottenere informazioni più precise sulle traslazioni ha due categorie di dispositivi hardware: quattro sensori da collocare nel luogo di lavoro e trasmettitori che deve indossare l'operatore (figura 21). Il trasmet-

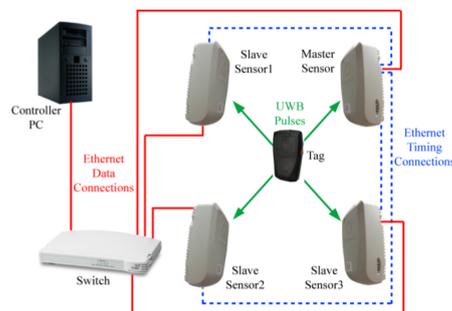


Figura 21: Il dispositivo Ubisense.

titore manda impulsi UWB ai sensori che tramite tecniche di *TDOA*

(Time-Difference of Arrival) e AOA (Angle of Arrival) [37] stimano la posizione globale in x , y , z dell'operatore. I sensori sono collegati a uno switch Ethernet e inviano i dati a un pc che accede ad essi tramite una libreria software. In [37] sono riportati i pregi del dispositivo Ubisense: immunità al multipath fading, immunità all'interferenza con le onde radio, dimensioni contenute dell'architettura hardware, nessuna occlusione.

Il sistema Ubisense campiona a basse frequenze: 5-9 Hz, mentre il GypsyGyro-18 fornisce dati a frequenze più elevate 30-120 Hz ma accumulando errori. La soluzione proposta in [37] è di combinare le misure sulla traslazione globale dei due sistemi, poiché il GypsyGyro-18 fornirà dati ad alta frequenza e il sistema Ubisense correggerà gli errori accumulati. Innanzitutto i dati forniti dai due sistemi (figura 22) vengono rappresentati secondo il sistema di riferimento del dispositivo Ubisense secondo la formula¹:

$$\begin{pmatrix} x^U \\ y^U \\ z^U \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 & x_G^U \\ -\sin \alpha & \cos \alpha & 0 & y_G^U \\ 0 & 0 & 1 & z_G^U \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x^G \\ y^G \\ z^G \\ 1 \end{pmatrix}. \quad (19)$$

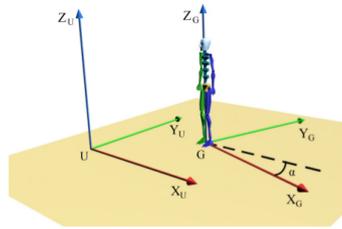


Figura 22: Sistemi di riferimento del GypsyGyro-18 e dell'Ubisense.

La presente e le successive formule sono riportate in [37], l'angolo α di figura 22 è un parametro conosciuto e costante: è l'angolo tra la direzione nord e l'asse Y_U del sistema Ubisense. Le coordinate delle traslazioni si ricavano con le seguenti formule:

$$x_G^U = x^U - x^G \cos \alpha - y^G \sin \alpha, \quad (20)$$

$$y_G^U = y^U + x^G \sin \alpha - y^G \cos \alpha, \quad (21)$$

$$z_G^U = z^U - z^G. \quad (22)$$

¹ In [37] viene fornita la matrice di cambiamento di coordinate del sistema di riferimento, assumendo che il sistema di riferimento del GypsyGyro-18 sia ruotato di un angolo α rispetto al sistema di riferimento Ubisense positivo, mentre in figura 22 tale angolo è negativo.

Infine le misure delle traslazioni del GypsyGyro-18 e dell'Ubisense vengono combinate con un algoritmo basato sul filtro di Kalman descritto in [37].

In sintesi, in [37] sono usati due dispositivi per monitorare l'operatore: il GypsyGyro-18 di Animazoo e un sistema di localizzazione a UWB di Ubisense. Questa scelta è dovuta al fatto che il GypsyGyro-18 misura le posizioni angolari in modo molto accurato, ha una frequenza di campionamento sufficientemente elevata, ma la stima delle traslazioni non è affidabile. Per avere informazioni più precise sulla posizione assoluta dell'operatore viene usato un sistema di localizzazione di Ubisense, che però ha una frequenza di campionamento bassa. Per avere la stima finale sulla posizione globale dell'operatore viene usato un algoritmo basato sul filtro di Kalman.

SCHEMI PER LA COLLABORAZIONE TRA UOMO E ROBOT

ARCHITETTURE DI ALTO LIVELLO PER COMPITI COLLABORATIVI

In questa sezione verranno affrontate tre architetture di alto livello, cioè schemi generali da cui si parte per progettare il software e l'hardware necessari al sistema di controllo del robot, per realizzare le operazioni collaborative.

Un'architettura basata su ROS

Un'altra possibilità che prevede un framework con middleware ROS (*Robot Operating System* [38]) è descritta in [32]. Essa si basa su strumenti di *Off-Line Programming* (OLP) per ridurre i tempi di programmazione online e di tracciamento del percorso. L'idea è che la coordinazione tra uomo e robot sia possibile senza usare moduli software specifici per la generazione di programmi per il robot.

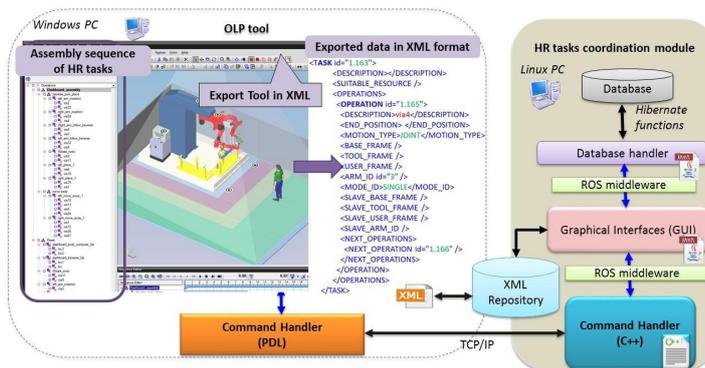


Figura 23: Architettura basata su ROS.

I moduli principali visibili nella figura sono due: *OLP tool*, che è usato per la simulazione dell'assemblaggio, e *HR task coordination*, per la coordinazione tra uomo e robot attraverso l'interfaccia grafica. I dati della simulazione del lavoro collaborativo sono esportati nel formato XML e immagazzinati nel blocco *XML Repository*, l'interfaccia grafica accede ai file XML per caricarli e depositarli nel database. Il database comunica con l'interfaccia tramite messaggi ROS e l'interfaccia accede a un file *HR-task data*, che contiene le istruzioni del lavoro collaborativo, attraverso il *Database handler*. I dati sono mandati al robot controller tramite il *command handler* e il protocollo di comunicazione *TCP/IP*, tutte le informazioni sui file *HR-tasks* sono

inviata al robot controller attraverso l'interazione dei due *command handler* uno dal lato del robot e l'altro della HR task coordination.

Il programma contenuto nel file XML è un modello gerarchico basato su tre categorie: *order*, *task* e *operation*. La prima categoria, *order* è la radice dell'albero che si dirama in *tasks* e *operations* nell'unità grafica, per mostrare all'operatore quali compiti dovranno essere svolti. La categoria *task* contiene un certo numero di *operation* che possono essere eseguite dal robot, dall'uomo o da entrambi. La terza categoria *operation* se riferita all'uomo comprende un certo numero di passi da eseguire senza argomenti, mentre nel caso del robot gli argomenti, come per esempio la posizione, sono compresi. Quando un *task* è assegnato all'operatore, questo è avvisato da una finestra e deve selezionare *start* per iniziare il compito e *stop* una volta terminato in modo che successivamente il robot possa riprendere a lavorare in modalità automatica.

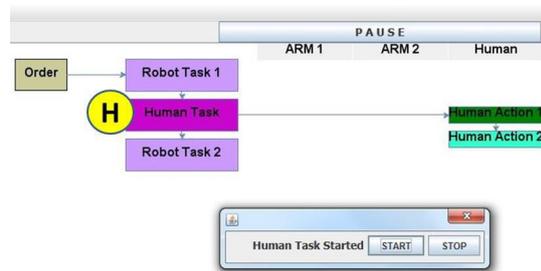


Figura 24: Finestra che abilita l'operatore a eseguire e terminare l'operazione collaborativa.

L'uso di OLP tools permette all'utente di simulare e progettare una cella industriale prima del setup in produzione, inoltre i file esportati possono essere usati senza che debbano essere tradotti nel linguaggio di programmazione del controller del robot, infine tutti i dati sono immagazzinati in database esterni in modo che siano accessibili anche da distante per modificarli e verificarli.

Un'architettura basata sul framework MAROCO e su DESCRIPTION LOGIC

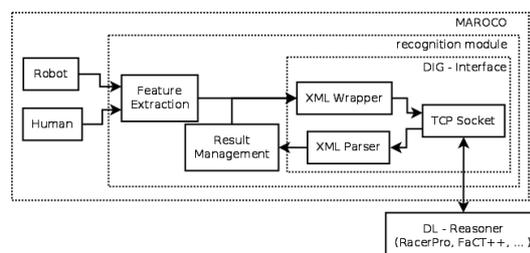


Figura 25: Componenti del modulo di riconoscimento.

In [39] è presentato un metodo che combina *Description Logics* (DLs), cioè logiche descrittive usate come mezzo per rappresentare la conoscenza e per inferire su attività, situazioni e aspettative, con il framework *MAROCO*, che analizza le immagini, stimando parametri cinematici come l'orientamento della testa, della parte superiore del corpo, la posizione delle braccia e fa l'analisi del rischio.

Il modulo di riconoscimento, presentato in figura 25, stabilisce la comunicazione con il ragionatore di DLs tramite la *DIG-interface*. La *DIG-interface* segue l'approccio *tell&ask*: la definizione della conoscenza di base è data dalle operazioni *tell*, i risultati del ragionatore possono essere recuperati tramite le operazioni *ask* [40]. La conoscenza di base è divisa nella conoscenza generale e in quella sugli individui nel dominio. La conoscenza sugli individui nel dominio è creata ad ogni ciclo dal *recognition module* e dipende sia dai parametri cinematici dell'operatore: gli angoli di entrambi i gomiti, quelli tra le braccia e le spalle rispetto all'asse superiore, quello che risulta dalla differenza tra l'orientamento della testa e il robot, la velocità e l'attrezzo usato; che da quelli del robot: lo stato del movimento e quello della pinza. Il modulo *feature extraction* attribuisce ad ogni ciclo un valore ai parametri cinematici dell'operatore e a quelli del robot, popolando la conoscenza di base con stringhe che vengono valutate dal ragionatore. Tutte le azioni riconosciute sono immagazzinate dal *result management* e sono recuperate durante una nuova creazione della conoscenza di base.

L'architettura JAHIR

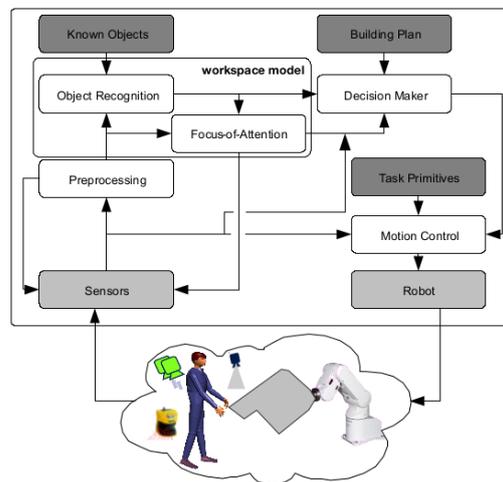


Figura 26: Schema dell'architettura di alto livello JAHIR.

In team la coordinazione richiede ai partecipanti di pianificare e eseguire le loro azioni in base a ciò che anticipano dagli altri membri, affinché la collaborazione tra uomo e robot sia efficiente va realizzata questa condizione. Un sistema cognitivo percepisce l'ambiente trami-

te sensori, processa il loro input e reagisce a situazioni basate sulla conoscenza in modo appropriato. Questo risultato è raggiunto chiudendo l'anello che comprende percezione, cognizione e azione. Un esempio di quanto scritto è il progetto *JAHIR (Joint Action for Humans and Industrial Robots)* [41], in cui l'architettura di alto livello è divisa in tre moduli: *joint attention*, *action observation* e *task sharing and action coordination*. Il primo modulo si basa sul fatto che il sistema deve saper riconoscere dove si sta concentrando l'attenzione dell'operatore, quello che viene definito *focus-of-attention* (vedi figura 26). Esso può essere ricavato tramite telecamere con tecniche di tracciamento o da data gloves; il sistema deve riconoscere il fatto che l'operatore stia puntando a qualcosa, calcolare la posizione indicata dall'operatore e poi trasformare questa informazione per stimare il focus-of-attention; anche la posizione della testa dell'operatore può essere usata per tale scopo. Questo passaggio è importante perché uomo e robot si focalizzano sugli stessi oggetti, i dati raccolti sono salvati nel database *known objects*. Per riconoscere gli oggetti è usato un approccio basato sui template nel modulo *Object Recognition*.

Il secondo modulo *Task Sharing* si basa sul fatto che per raggiungere un determinato obiettivo, per esempio l'assemblaggio di un prodotto, il robot deve conoscere il processo e i compiti dell'operatore, inoltre la rappresentazione del processo dovrebbe essere la più generica possibile per cambiare il ruolo dinamicamente anche durante la produzione. La conoscenza degli step e delle capacità primitive è rappresentata dal database *Building Plan*.

Il terzo modulo è *Action Observation and Coordination*, tutte le informazioni percepite dai sensori formano una rappresentazione interna dello spazio di lavoro, il *workspace model*. Il workspace model potrebbe anche contenere informazioni dell'inventario, per esempio quanti pezzi sono stoccati, dove possono essere piazzati e quale pezzo è sulla linea di assemblaggio. Queste informazioni assieme a quelle del Building Plan Module, dei sensori e del focus-of-attention sono usate dal *Decision Maker* per stabilire la prossima azione. Per controllare il movimento del robot in tempo reale, in modo da evitare collisioni con l'operatore, i sensori sono collegati al blocco *Motion Control*. Per ragioni di sicurezza la programmazione dei movimenti del robot non può essere off-line.

Poiché il blocco computazionale che processa le immagini richiede molta potenza di calcolo, i moduli del sistema sono divisi tra più computer e la comunicazione tra questi moduli avviene tramite il *middleware ICE-Internet Communication Engine*. Il sistema deve conoscere, o essere in grado di imparare, il piano di assemblaggio; altre informazioni importanti da archiviare in un database sono: quali strumenti sono disponibili al robot e quali oggetti possono essere usati dal robot. Per i contenitori riconosciuti sul banco di lavoro dal sistema è noto il colore, il contenuto, la posizione secondo le coordinate x

e y , il numero di parti rimanenti in essi.

Il sistema osserva le azioni dell'uomo nello spazio di lavoro collaborativo tramite sensori e le usa, assieme alla conoscenza del piano di assemblaggio, per anticipare il prossimo compito dell'operatore in modo da comportarsi in un certo modo, per esempio fornire un attrezzo, in base all'oggetto prelevato. In [41] sono impiegate quattro telecamere (vedi figura 27), una è montata vicino al *tool center point* (punto definito per una certa applicazione, rappresentato secondo le coordinate del sistema di riferimento dell'interfaccia meccanica [4]) del manipolatore e può essere usata per rilevare oggetti noti e la loro posizione, in modo che il robot svolga con successo il compito di passare all'operatore lo strumento giusto. Altre due telecamere monitorano lo spazio di lavoro collaborativo da sinistra e da destra rispettivamente per stimare la posa dell'operatore e la posizione delle sue mani. La terza telecamera è diretta frontalmente per tracciare la posizione e la rotazione della testa dell'operatore in modo da stimare il focus-of-attention. La telecamera frontale può essere anche usata

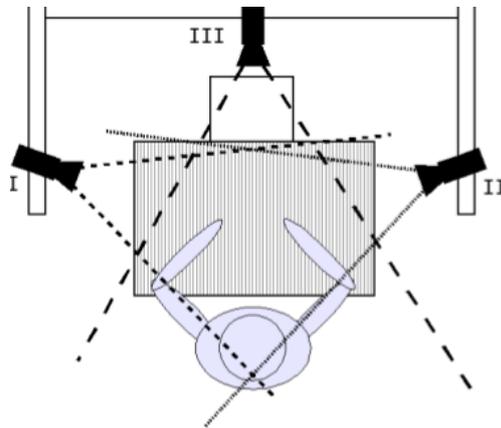


Figura 27: Posizionamento delle telecamere nella configurazione sperimentale.

per riconoscere la faccia dell'operatore, in modo da tarare il sistema affinché offra più supporto a un lavoratore meno esperto, o di fornire in un altro modo l'oggetto a un lavoratore mancino. Per identificare gesti come puntare a qualcosa o chiudere la mano a pugno è disponibile il *P5 Dataglove* (figura 28) che rileva le coordinate della posizione della mano e le dita piegate. Uno svantaggio di questo sistema è la scarsa accuratezza che è approssimativamente da 10 a 20 mm e la mano deve essere a meno di 1 m dalla torre del sensore per essere rilevata. Poiché è richiesta maggiore accuratezza per stimare la posizione della mano quando il robot deve passare all'operatore un attrezzo, in [41] è suggerito anche l'uso di un sistema stereo di tracciamento a infrarossi con accuratezza di 1 mm.



Figura 28: Il P5 data glove.

TECNICHE DI APPRENDIMENTO AUTOMATICO

Le tecniche di apprendimento automatico consentono ai robot di apprendere una sequenza di azioni senza il compito di un programmatore, osservando un modello: un operatore, un altro robot o un operatore/robot virtuale. È stato proposto di trasferire tecniche di apprendimento automatico, quali *LfD*, in ambienti virtuali per ridurre il tempo, il lavoro e il costo dello sviluppo dei metodi nel mondo reale [42]. Quindi un modello virtuale dell'uomo, il *virtual human*, è controllato dall'operatore perché ne riproduca il comportamento in termini di azioni, un altro agente, il *virtual learner*, osserva le azioni dell'operatore virtuale per apprenderle. Questa prima fase si chiama *behavioural learning* [42], nella seconda fase *collaborative learning* [42], il *virtual learner* impara a collaborare con il *virtual human*. Infine quando il processo di apprendimento si è concluso nell'ambiente virtuale, i risultati possono essere trasferiti su un vero robot, oppure su un software, ciò è a vantaggio della sicurezza, infatti solo i comportamenti ritenuti sicuri nell'ambiente virtuale saranno implementati nel mondo reale [42].

I robot che imparano da dimostrazioni sono sistemi adattativi [42], poiché sono progettati per modificare il loro comportamento durante i cicli di lavoro. Le modalità di certificazione dei sistemi adattativi sono diverse rispetto a quelle dei sistemi tradizionali di cui si conosce il comportamento dal progetto, infatti esse dipendono dall'applicazione in azienda del sistema adattativo oltre che dal progetto. Una parte della certificazione viene fatta quando il robot è operativo, secondo l'approccio *just-in-time certification* [43]: se il comportamento di un sistema è fissato durante il progetto, la valutazione della sicurezza si basa sul comportamento predefinito, bisogna considerare se le caratteristiche del sistema seguono un insieme prestabilito di requisiti di sicurezza. Se il procedimento di verifica di tali requisiti può essere automatizzato, allora è possibile applicarlo in azienda durante l'uso del sistema adattativo. Ciò comporta la codifica degli standard di sicurezza in una procedura automatica applicabile nel luogo di

lavoro e capace di prevenire ogni comportamento che ne causi la violazione [42]. Anche l'approccio *learning from demonstration* (LfD) può essere applicato nel luogo di lavoro con la limitazione ai soli risultati di apprendimento sicuri. Tale approccio si basa sul fatto che per eseguire un compito, una sequenza di azioni è applicata a un dato *world state*. Un *world state* consiste di stati S e azioni A , che spaziano dalle *low level motion*, alle *high levels behaviors* [44], ciascuna azione trasforma il *world state*, l'ultima azione dovrebbe risultare in uno stato che riflette l'esecuzione del compito. Dopo aver osservato l'operazione, il robot è guidato in un processo di apprendimento tramite lo sviluppo di una *policy*, cioè una mappa che collega *world state* e azioni: essa permette al robot di scegliere un'azione in base all'attuale *world state*. Attualmente lo sviluppo di una *policy* è un compito ristretto ai soli esperti di robotica, tuttavia LfD non richiede conoscenze approfondite sulla dinamica, ciò consentirà anche ai meno esperti di dedicarsi [44]. Formalmente, la funzione di transizione probabilistica che mappa stati e azioni è $T(s'|s, a) : S \times A \times S \rightarrow [0, 1]$, si assume che lo stato S non sia totalmente osservabile, il robot che osserva per imparare ha accesso a uno stato Z attraverso l'applicazione $M : S \rightarrow Z$, una *policy* $\pi : Z \rightarrow A$ seleziona le azioni basate sulle osservazioni del *world state*. Gli stati e le azioni sono registrate grazie alla dimostrazione d_j di un insegnante, con $d_j \in D$ composta da k_j coppie di osservazioni e azioni: $d_j = (z_j^i, a_j^i)$, $z_j^i \in Z$, $a_j^i \in A$, $i = 0 \dots k_j$ [44]. Il robot che ha ricevuto la dimostrazione deriva una *policy* che gli permette di riprodurre l'operazione desiderata, ai fini dell'apprendimento sono importanti due parametri: chi controlla la dimostrazione, chi esegue la dimostrazione. La dimostrazione può essere eseguita autonomamente da un robot, in tal caso il robot ricopre i due parametri sopra indicati, oppure l'operazione può essere eseguita da un robot che viene teleoperato dall'uomo, in tal caso il robot esegue la dimostrazione e l'uomo controlla la dimostrazione. I dati possono essere acquisiti in blocco al termine della dimostrazione *batch learning* o progressivamente durante la dimostrazione. Lo sviluppatore deve scegliere se operare in un dominio discreto o continuo: nel caso di un robot che deve raccogliere un pezzo, trasportarlo e ricollocarlo, lo scenario in cui il robot sta trasportando il pezzo e quello in cui il pezzo occupa una posizione prestabilita può essere rappresentato da un valore booleano, oppure dalle posizioni in uno spazio tridimensionale dell'end-effector e dell'oggetto. Quando si sceglie un algoritmo per generare una *policy* sono importanti due fattori: l'approccio adottato per derivare la *policy* e se la performance può migliorare dopo l'insegnamento. Ci sono tre approcci per derivare una *policy* dai dati della dimostrazione: *mapping function*, *system model*, *plans* [44], illustrati in figura 29. Secondo la modalità *mapping function* i dati della dimostrazione sono usati per approssimare l'applicazione che correla gli stati di osservazione del robot alle azioni $f() : Z \rightarrow A$. Nella modalità

system model i dati della dimostrazione sono usati per determinare un modello dinamico $T(s'|s, a)$ e una *reward function* $R(s)$. Una policy è derivata successivamente usando queste informazioni. Per quanto riguarda l'ultima modalità, plans, i dati della dimostrazione e altre informazioni sull'utente sono usate per imparare regole che associano un insieme di pre- e post-condizioni per ciascuna azione e, se possibile, anche un modello dinamico $T(s'|s, a)$. Una sequenza di azioni viene successivamente pianificata usando queste informazioni. Come esempi, nel caso mapping function $f() : Z \rightarrow A$, Z può essere la posizione secondo le tre dimensioni dell'end-effector, A la velocità desiderata dell'end-effector e $f()$ è la funzione da apprendere che collega Z ad A . Secondo l'approccio system model, invece, viene imparato un modello di transizione $T(s'|s, a)$, per esempio quando il robot compie l'azione di prelevare un pezzo nel momento in cui il pezzo è sul tavolo, lo stato finale risulta essere il pezzo prelevato dal robot. Usando questo modello, la policy derivata indica la migliore azione da eseguire in un dato stato per raggiungere l'obiettivo. Per quanto riguarda l'ultimo metodo, usando il *planning approach*, le pre- e post-condizioni riguardo l'esecuzione di un'azione $L(\{preC, postC\}|a)$ vengono imparate dalle dimostrazioni. Per esempio, l'azione di prelevare un pezzo richiede che il pezzo sia sul tavolo come pre-condizione. Informazioni di questo tipo vengono usate per generare una sequenza di azioni che terminano col raggiungimento dello stato desiderato. L'appren-

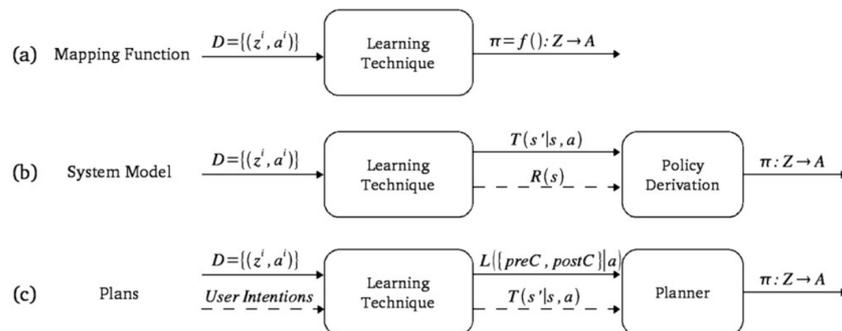


Figura 29: Tecniche di derivazione della policy.

dimento del robot si basa sulla dimostrazione che dipende da chi la esegue: in molti casi le capacità dell'operatore sono inferiori rispetto a quelle del robot, per esempio il robot è molto più veloce e accurato dell'uomo nell'eseguire le azioni, poiché la policy è derivata da questi esempi le abilità del robot saranno limitate da quelle dell'uomo.

L'insieme di dati della dimostrazione è composto da coppie stato azione registrate durante la riproduzione del comportamento desiderato da parte dell'insegnante, a seconda dell'approccio variano la piattaforma usata dall'insegnante per eseguire la dimostrazione e la modalità che consente di raccogliere i dati sulla dimostrazione. Esempi comprendono i casi dei sensori posizionati sul robot che registrano

le azioni comandate attraverso *teach-pendant*: unità da tenere in mano collegata al sistema di controllo del robot mediante la quale il robot può essere mosso o programmato [4], dall'operatore e quelli che si basano sulle telecamere che registrano i movimenti dell'operatore. Perché il metodo LfD sia efficace gli stati e le azioni che fanno parte del dataset devono essere fruibili dal robot. Nel caso più diretto gli stati e le azioni di chi esegue la dimostrazione vengono mappati direttamente nel robot. In realtà ciò non è quasi mai possibile poiché il robot e l'operatore differiscono per organi di senso e per la struttura: la telecamera di un robot non rileva l'ambiente esterno allo stesso modo della vista dell'uomo, la pinza del robot non applica la forza allo stesso modo delle mani dell'operatore. Il problema della corrispondenza si riferisce al fatto di trovare una mappa tra chi esegue la dimostrazione e il robot che permetta il trasferimento di informazioni da uno all'altro. La *Record Mapping* riferisce se stati/azioni visitati dal-

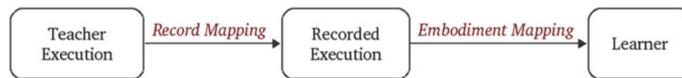


Figura 30: Processo in cui viene mappata la dimostrazione dall'insegnante al soggetto che la deve apprendere.

l'insegnante durante la dimostrazione sono stati registrati nel dataset. L'*embodiment mapping* riferisce se gli stati/azioni registrate nel dataset sono esattamente quelli che il robot dovrebbe osservare/eseguire. Quando la record mapping è l'identità $I(z, a)$ gli stati/azioni dell'insegnante sono registrati direttamente nel dataset. Negli altri casi gli stati/azioni dell'insegnante sono codificati come $g_R(z, a) \neq I(z, a)$ e registrati nel dataset. Analogamente quando l'*embodiment mapping* è l'identità $I(z, a)$ gli stati/azioni vengono mappati direttamente nel robot dal dataset, altrimenti l'*embodiment mapping* consiste in una funzione $g_E(z, a) \neq I(z, a)$. Un esempio di $g_R(z, a) \neq I(z, a)$ record mapping $A_T \rightarrow D$ è dato da un operatore che insegna a un robot a raccogliere un oggetto, il robot osserva le azioni dell'uomo A_T , cioè gli angoli tra gli arti, mediante una telecamera e li riproduce tra i suoi membri con le azioni A_L . In tal caso le azioni dell'uomo sono sconosciute al robot e vengono estratte dalle immagini della telecamera. Inoltre il physical embodiment dell'operatore è diverso da quello del robot e le azioni A_T sono diverse da quelle A_L , per rendere la dimostrazione comprensibile al robot deve essere applicata la mappa $D \rightarrow A_L$, questo è un esempio di $g_E(z, a) \neq I(z, a)$ embodiment mapping.

L'acquisizione dati LfD si suddivide in due categorie: *Demonstration* e *Imitation* [44]. Nel caso demonstration non serve l'*embodiment mapping*, perché le azioni vengono eseguite dal robot, quindi $g_E(z, a) \equiv I(z, a)$. Invece per quanto riguarda l'*imitation* esiste l'*embodiment mapping*, perché la dimostrazione è eseguita su una piattaforma che

non è il robot, così $g_E(z, a) \neq I(z, a)$. Si possono fare ulteriori distinzioni in base al record mapping, cioè come i dati della dimostrazione vengono acquisiti. Quando le operazioni eseguite dall'uomo vengono dimostrate non serve l'embodiment mapping, questa situazione è presentata nella colonna di sinistra della figura 31. Il record map-

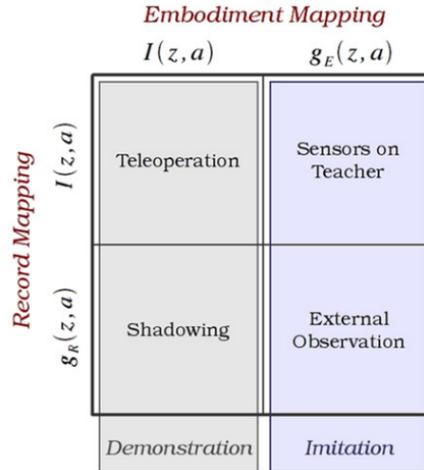


Figura 31: Demonstration e imitation, identity e non-identity per embodiment mapping e record mapping.

ping potrebbe non essere diretto per stati/azioni se la registrazione di stati/azioni viene ricavata dall'analisi dei dati. Ci sono due approcci per fornire i dati della dimostrazione al robot: *teleoperation* e *shadowing* [44]. Nel caso teleoperation l'operatore guida il robot tramite teach pendant e i sensori del robot memorizzano l'esecuzione; il record mapping è diretto $g_R(z, a) \equiv I(z, a)$. Per il secondo caso, shadowing, l'operatore esegue il compito, il robot lo imita e contemporaneamente tramite i dati dei sensori del robot viene memorizzata l'esecuzione; il record-mapping non è diretto: $g_R(z, a) \neq I(z, a)$. Invece, poiché i dati memorizzati provengono dai sensori del robot l'embodiment mapping è diretto $g_E(z, a) \equiv I(z, a)$.

Imitation è nella colonna di destra della figura 31, non c'è l'identità per l'embodiment mapping $g_E(z, a) \neq I(z, a)$, ci sono anche in questo caso due approcci per fornire i dati a seconda che il record mapping sia o meno un'identità: *sensors on teacher* e *external observation* [44]. Nel caso sensors on teacher i sensori sono posizionati sul corpo dell'operatore, o della piattaforma che esegue la dimostrazione, e i dati registrati dai sensori durante l'esecuzione vengono memorizzati; in tal caso si ha un'identità per il record mapping $g_R(z, a) = I(z, a)$. Per quanto riguarda l'approccio external observation i sensori sono posizionati esternamente, non sul corpo dell'operatore, in tal caso il record mapping non è diretto $g_R(z, a) \neq I(z, a)$. Il primo metodo è più oneroso in termini economici perché richiede sensori particolari: sensori collocati su un tessuto perché siano indossabili dall'operatore,

e un ambiente attrezzato: stanze provviste di telecamere, mentre il secondo è meno costoso ma anche meno preciso. Infatti il robot non registra direttamente gli stati/azioni dell'operatore, ma essi devono essere ricavati dai dati, comportando così un'incertezza tra gli stati/azioni originali e quelli dedotti. Un pregio di questo metodo è di non richiedere sensori specifici e ambiente attrezzato.

CONCLUSIONI

Riguardo al tema della collaborazione tra uomo e robot, in questa tesi è stato riportato, anzitutto, il contesto in cui tale modalità operativa si inserisce, assieme a una panoramica generale degli altri obiettivi della quarta rivoluzione industriale. Sempre nell'introduzione è stato riassunto il percorso storico della normativa sul rapporto tra uomo e robot in azienda; inoltre, è stato mostrato uno schema che mostra un esempio di hardware per le operazioni collaborative: sensori visivi, di prossimità, dispositivi alla realtà aumentata indossabili dall'operatore. Successivamente, sono state riportate le modalità collaborative descritte nella normativa; sono stati trattati alcuni esempi sul tema delle misure di sicurezza per l'implementazione della collaborazione tra uomo e robot in ambiente industriale, sia nel caso in cui il robot sia fisso in un determinato spazio, che in quello in cui il robot sia mobile su piattaforma; sono stati sintetizzati alcuni sistemi che consentono di monitorare l'uomo nello spazio di lavoro. Infine sono state riportate alcune architetture di alto livello e si è accennato alle tecniche di apprendimento automatico che consentono al robot di imparare le operazioni collaborative tramite l'insegnamento di un operatore. L'obiettivo di questa tesi è quello di fare una piccola indagine sulla collaborazione tra uomo e robot industriali attraverso parte del materiale pubblicato dai ricercatori, poiché il tema trattato è attuale e di interesse industriale per i numerosi vantaggi che offre: velocità, precisione, forza del robot assieme alle capacità cognitive dell'uomo e quindi maggiore flessibilità.

BIBLIOGRAFIA

- [1] Michael Rüßmann, Markus Lorenz, Philipp Gerbert, Manuela Waldner, Jan Justus, Pascal Engel, and Michael Harnisch. Industry 4.0: The future of productivity and growth in manufacturing industries. *Boston Consulting Group*, page 14, 2015.
- [2] David Romero, Peter Bernus, Ovidiu Noran, Johan Stahre, and Åsa Fast-Berglund. The operator 4.0: Human cyber-physical systems & adaptive automation towards human-automation symbiosis work systems. In *APMS (Advances in Production Management Systems)*, 2016.
- [3] Jeff Fryman and Bjoern Matthias. Safety of industrial robots: From conventional to collaborative applications. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–5. VDE, 2012.
- [4] ISO 10218. *ISO 10218 Robots and robotic devices – Safety requirements for industrial robots, with parts 1 (Robots) and 2 (Robot systems and integration)*, ISO Copyright Office, Geneva, 2011.
- [5] Rosemarie Velik, S Yahjanejad, B Dieber, Mathias Brandstötter, G Paar, Lucas Paletta, H Zeiner, Patrick Luley, F Fuhrmann, and M Hofbauer. A step forward in human-robot collaboration—the project collrob. 1st OAGM-ARW Joint Workshop, Wels Austria, 2015.
- [6] Carlos Morato, Krishnanand N Kaipa, Boxuan Zhao, and Satyandra K Gupta. Toward safe human robot collaboration by using multiple kinects based real-time human tracking. *Journal of Computing and Information Science in Engineering*, 14(1), 2014.
- [7] Martin Zirkl, Anurak Sawatdee, Uta Helbig, Markus Krause, Gregor Scheipl, Elke Kraker, Peter Andersson Ersman, David Nilsson, Duncan Platt, Peter Bodö, et al. An all-printed ferroelectric active matrix sensor network based on only five functional materials forming a touchless control interface. *Advanced Materials*, 23(18):2069–2074, 2011.
- [8] Chien-Ming Huang and Bilge Mutlu. Anticipatory robot control for efficient human-robot collaboration. In *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*, pages 83–90. IEEE, 2016.
- [9] Patrik Polatsek, Wanda Benesova, Lucas Paletta, and Roland Perko. Novelty-based spatiotemporal saliency detection for predic-

- tion of gaze in egocentric video. *IEEE Signal Processing Letters*, 23(3):394–398, 2016.
- [10] Lucas Paletta, Katrin Santner, Gerald Fritz, Heinz Mayer, and Johann Schrammel. 3d attention: measurement of visual saliency using eye tracking glasses. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 199–204. ACM, 2013.
- [11] Simon Keizer, Mary Ellen Foster, Andre Gaschler, Manuel Giuliani, Amy Isard, and Oliver Lemon. Handling uncertain input in multi-user human-robot interaction. In *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, pages 312–317. IEEE, 2014.
- [12] Guy Hoffman and Cynthia Breazeal. Collaboration in human-robot teams. In *AIAA 1st Intelligent Systems Technical Conference*, page 6434, 2004.
- [13] Marco Ewerton, Gerhard Neumann, Rudolf Lioutikov, Heni Ben Amor, Jan Peters, and Guilherme Maeda. Learning multiple collaborative tasks with a mixture of interaction primitives. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1535–1542. IEEE, 2015.
- [14] Christopher D Wickens, Sallie E Gordon, Yili Liu, and John Lee. An introduction to human factors engineering. 1998. Longman, New York.
- [15] Lorenza Tiberio, Amedeo Cesta, and Marta Olivetti Belardinelli. Psychophysiological methods to evaluate user's response in human robot interaction: a review and feasibility study. *Robotics*, 2(2):92–121, 2013.
- [16] Maurizio Garbarino, Matteo Lai, Dan Bender, Rosalind W Picard, and Simone Tognetti. Empatica E3—A wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition. In *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on*, pages 39–42. IEEE, 2014.
- [17] Arindam Ghosh, Juan Manuel Mayor Torres, Morena Danelli, and Giuseppe Riccardi. Detection of essential hypertension with physiological signals from wearable devices. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 8095–8098. IEEE, 2015.
- [18] Brandy Warwick, Nicholas Symons, Xiao Chen, and Kaiqi Xiong. Detecting driver drowsiness using wireless wearables. In *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on*, pages 585–588. IEEE, 2015.

- [19] Chatchavan Wacharamanotham, Jonathan Meyer, Jonathan Diehl, and Jan Borchers. The interactive bracelet: An input device for bimanual interaction. In *Mobile HCI 2010 Workshop on Ensembles of On-Body Devices, Lisbon, Portugal*, 2010.
- [20] Asilbek Ganiev, Ho-Sun Shin, and Kang-Hee Lee. Study on virtual control of a robotic arm via a myo armband for the selfmanipulation of a hand amputee. *Int. J. Appl. Eng. Res*, 11(2):775–782, 2016.
- [21] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics*, 43(5):1318–1334, 2013.
- [22] Cha Zhang and Zhengyou Zhang. Calibration between depth and color sensors for commodity depth cameras. In *Computer Vision and Machine Learning with RGB-D Sensors*, pages 47–64. Springer, 2014.
- [23] Daniel Herrera, Juho Kannala, and Janne Heikkilä. Accurate and practical calibration of a depth and color camera pair. In *International Conference on Computer analysis of images and patterns*, pages 437–445. Springer, 2011.
- [24] David Fiedler and Heinrich Müller. Impact of thermal and environmental conditions on the kinect sensor. In *Advances in Depth Image Analysis and Applications*, pages 21–31. Springer, 2013.
- [25] ISO 15066. *Robots and robotic devices – Collaborative robots*.
- [26] IEC 60204-1. *IEC 60204-1: (2005), Safety of machinery- Electrical equipment of industrial machines - Part 1: General requirements*.
- [27] ISO 13855. *ISO 13855:2002, Safety of machinery – Positioning of protective equipment with respect to the approach speeds of parts of the human body, (2002)*.
- [28] JA Corrales, GJ Garcia Gomez, F Torres, and Véronique Perdereau. Cooperative tasks between humans and robots in industrial environments. *International Journal of Advanced Robotic Systems*, 9(3):94, 2012.
- [29] Vito Magnanimo, Steffen Walther, Luigi Tecchia, Ciro Natale, and Tim Guhl. Safeguarding a mobile manipulator using dynamic safety fields. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2972–2977. IEEE, 2016.
- [30] Robert Fisher, Simon Perkins, Ashley Walker, and Erik Wolfart. Hypermedia image processing reference. *Department of Artificial Intelligence, University of Edinburgh*, 1994.

- [31] Jack E Bresenham. Ambiguities in incremental line rastering. *IEEE Computer Graphics and Applications*, 7(5):31–43, 1987.
- [32] George Michalos, Sotiris Makris, Panagiota Tsarouchi, Toni Gua-sch, Dimitris Kontovrakis, and George Chryssolouris. Design considerations for safe human-robot collaborative workplaces. *Procedia CIRP*, 37:248–253, 2015.
- [33] Vittorio Rampa, Federico Vicentini, Stefano Savazzi, Nicola Pe-drocchi, Marcellso Ioppolo, and Matteo Giussani. Safe human-robot cooperation through sensor-less radio localization. In *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*, pages 683–689. IEEE, 2014.
- [34] Keith J O’Hara and Tucker R Balch. Pervasive sensor-less networks for cooperative multi-robot tasks. In *Distributed Autonomous Robotic Systems 6*, pages 305–314. Springer, 2007.
- [35] Stig Petersen and Simon Carlsen. Wirelesshart versus isa100. 11a: The format war hits the factory floor. *IEEE Industrial Electronics Magazine*, 5(4):23–34, 2011.
- [36] Florian Hecht, Pedram Azad, and Rudiger Dillmann. Marker-less human motion tracking with a flexible model and appearance learning. In *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*, pages 3173–3179. IEEE, 2009.
- [37] Juan Antonio Corrales, FA Candelas, and Fernando Torres. Hybrid tracking of human operators using imu/uwb data fusion by a kalman filter. In *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*, pages 193–200. IEEE, 2008.
- [38] Jit Ray Chowdhury. Ros: Robot operating system. *Autonomous Ground Vehicle Research Group, Indian Institute of Technology Kharagpur (AGV KGP), Kharagpur, India*, 2012.
- [39] Jürgen Graf, Stephan Puls, and Heinz Wörn. Recognition and understanding situations and activities with description logics for safe human-robot cooperation. In *Proceedings of the 2nd International Conference on Advanced Cognitive Technologies and Applications: Cognitive 2010*, pages 90–96. Citeseer, 2010.
- [40] Stephan Puls, Heinz Wörn, and Jürgen Graf. *Cognitive robotics in industrial environments*. INTECH Open Access Publisher, 2012.
- [41] Claus Lenz, Suraj Nair, Markus Rickert, Alois Knoll, Wolfgang Rosel, Jurgen Gast, Alexander Bannat, and Frank Wallhoff. Joint-action for humans and industrial robots for assembly tasks. In *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, pages 130–135. IEEE, 2008.

- [42] Kerstin Eder, Chris Harper, and Ute Leonards. Towards the safety of human-in-the-loop robotics: Challenges and opportunities for safety assurance of robotic co-workers'. In *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, pages 660–665. IEEE, 2014.
- [43] Brian Randell and John Rushby. Distributed secure systems: Then and now. In *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual*, pages 177–199. IEEE, 2007.
- [44] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

