

Articolo originale in lingua inglese. Ricevuto il 9 settembre 2020, accettato il 21 settembre 2020, pubblicato il 23 settembre 2020. Versione tradotta datata 5 ottobre 2020.

Per favore citare come A. de Giorgio and L. Wang, "Artificial Intelligence Control in 4D Cylindrical Space for Industrial Robotic Applications", in IEEE Access, vol. 8, pp. 174833-174844, 2020, doi: [10.1109/ACCESS.2020.3026193](https://doi.org/10.1109/ACCESS.2020.3026193).

# Controllo ad intelligenza artificiale in spazi cilindrici 4D per applicazioni di robotica industriale

ANDREA de GIORGIO<sup>1</sup> e LIHUI WANG<sup>1</sup>

<sup>1</sup>Dipartimento di Ing. della Produzione, Scuole di Ing. Industriale e Management, KTH Istituto Reale di Tecnologia, 114 28 Stoccolma, Svezia

Per corrispondenza: Andrea de Giorgio (e-mail: [andreadg@kth.se](mailto:andreadg@kth.se)). Traduzione in italiano a cura di Andrea de Giorgio.

**ABSTRACT** Questo articolo scientifico sostiene la tesi che un algoritmo di controllo, facente uso di intelligenza artificiale, per poter essere efficiente necessita di sfruttare le simmetrie integrate in ogni manipolatore robotico industriale così che quest'ultimo possa essere ulteriormente caratterizzato ed utilizzato. Il prodotto di questo miglioramento è uno spazio discreto a griglia cilindrica quadridimensionale (4D) che può direttamente sostituire complessi modelli robotici.  $A^*$  è scelto per il suo ampio utilizzo tra simili algoritmi di ricerca, in modo da studiare i vantaggi e gli svantaggi del controllo di robot industriali tramite la griglia discreta cilindrica 4D. Lo studio mostra che questo approccio consente di controllare un robot senza alcuna conoscenza specifica dei modelli cinematici e dinamici del robot al momento della pianificazione e dell'esecuzione. In effetti, le posizioni dei giunti del robot per ciascuna cella della griglia vengono precalcolate e memorizzate come conoscenza, quindi recuperate rapidamente dall'algoritmo di ricerca di percorso quando necessario. Lo spazio discreto cilindrico 4D presenta sia i vantaggi dello spazio di configurazione che dello spazio di lavoro cartesiano tridimensionale del robot. Poiché l'ottimizzazione del percorso è il nucleo di qualsiasi algoritmo di ricerca, incluso  $A^*$ , la griglia cilindrica 4D fornisce uno spazio di ricerca che può incorporare ulteriori conoscenze sotto forma di proprietà delle celle, inclusa la presenza di ostacoli e l'occupazione volumetrica dell'intero corpo del robot industriale, da usare in applicazioni per l'evitamento degli ostacoli. Il compromesso principale è tra una capacità limitata della conoscenza precalcolata nella griglia e la velocità di ricerca del percorso migliore. Questo approccio innovativo incoraggia l'uso di algoritmi di ricerca per applicazioni robotiche industriali, apre la via allo studio di altre simmetrie fisiche presenti in altri modelli di robot e pone le basi per l'applicazione di algoritmi dinamici per l'evitamento degli ostacoli.

**INDICIZZAZIONE** Robot industriale, controllo robotico, algoritmo, ricerca di percorso, intelligenza artificiale, simmetria cilindrica, spazio di ricerca, informazione volumetrica, evitamento degli ostacoli.

## I. INTRODUZIONE

Al di là dei molti benefici nel risolvere problemi in spazi multidimensionali, che ha portato all'uso di robot industriali con gradi di libertà ridondanti [1,2], la complessità degli algoritmi di intelligenza artificiale per la pianificazione di percorso o traiettoria cresce con la dimensionalità delle variabili di controllo, perché lo spazio di ricerca relativo diventa combinatoriamente più arduo da esplorare completamente [3]. L'applicazione degli algoritmi di intelligenza artificiale per la pianificazione di percorso o traiettoria riguarda sia i robot mobili che quelli industriali. La prima categoria riguarda i robot che possono muoversi

attraverso un dato ambiente, ad esempio terra, aria, acqua, etc. I robot industriali sono tipicamente caratterizzati da un braccio o multiple braccia mobili legate ad un corpo fisso e le parti mobili possono eventualmente collidere l'una con l'altra. I robot umanoidi sono l'unione di entrambi i casi, infatti di solito possono camminare in giro e usare le loro braccia. Nonostante le diverse sfide nel modellare tutti questi tipi di robot, gli algoritmi di controllo con l'intelligenza artificiale possono generalmente essere applicati a tutti i tipi di robot, con lievi differenze. Un robot mobile tipicamente si muove su superfici piate con ostacoli che possono essere mappati su una singola griglia bidimensionale (2D) [4]. Un

sistema di coordinate tridimensionale (3D) è necessariamente richiesto laddove i robot si muovano attraverso uno spazio 3D invece che su una superficie, rendendo i modelli della loro cinematica e dinamica più ardui da risolvere. Questo è il caso per alcune classi di robot mobili, di robot volanti come ad esempio i quadricopteri [5], e per la maggioranza dei manipolatori robotici industriali [6], i cui modelli devono rispondere della posizione di ciascuna delle loro parti in movimento.

La necessità di usare una terza dimensione per descrivere i movimenti dei robot industriali in spazio cartesiano è stata già sfidata da molti ricercatori che hanno introdotto controlli nello spazio delle configurazioni che è altamente multidimensionale [7–11]. Tuttavia, i vantaggi nell'uso dello spazio delle configurazioni sono un compromesso con la maggiore difficoltà di visualizzare ed operare in questo spazio, al punto che rappresentare altri oggetti interagenti con il robot nel suo spazio di configurazione diventa una disciplina di studio a sé stante [1,2,12–16].

Non è unusuale utilizzare algoritmi di pianificazione di percorso o traiettoria che siano basati su un compromesso tra lo spazio di configurazione e lo spazio di lavoro (cartesiano) come spazio di ricerca. Tra questi algoritmi val la pena citare i *rapidly-exploring random tree* {lett. alberi casuali ad esplorazione rapida} (RRT) [17] o le *probabilistic roadmaps* {lett. carte stradali probabilistiche} [18,19], nessuna delle quali è improntata a caratterizzare l'intero spazio robotico in anticipo in ragione dell'altissimo numero di configurazioni robotiche possibili e di posizioni spaziali, nonché la complessità in termini combinatoriali che deriva da essi. Questo articolo scientifico mostra che tale operazione è in effetti possibile in uno spazio di ricerca che sfrutti le simmetrie robotiche ed apre la via a possibili nuove versioni di questi algoritmi. Questo perché l'algoritmo proposto in questa pubblicazione non esegue una ricerca nello spazio di configurazione, bensì produce uno spazio a coordinate cilindriche grazie alle molte simmetrie cilindriche già presenti in quasi tutti i robot industriali. Oltre ciò, un'ulteriore dimensione è aggiunta allo spazio cilindrico per rappresentare gli orientamenti dell'estremità del robot {lett. end effector} (EE) in uno spazio cilindrico quadridimensionale (4D) ibrido. Le implicazioni geometriche delle configurazioni robotiche, come ad esempio le posizioni e gli orientamenti del polso o del gomito, sono riflesse nella struttura del sistema di coordinate cilindriche in 4D. Questo può essere considerato come una riduzione di dimensionalità dello spazio di configurazione robotica e porta sia i vantaggi del fare ricerca in uno spazio fisico a bassa dimensionalità che della più facile risoluzione dei problemi specifici relativi alle specifiche configurazioni robotiche, come ad esempio le singolarità [20].

L'articolo è strutturato come segue. Nella sezione II viene presentata della letteratura scientifica aggiuntiva. Nella sezione III le tipiche simmetrie cilindriche dei robot industriali sono spiegate e caratterizzate in forma di piani di raggiungibilità. Nella sezione IV i piani di raggiungibilità sono

acquisiti (A) e trasformati in uno spazio di ricerca cilindrico in 4D (B), ulteriori dati sono aggiunti alle celle come ad esempio l'informazione riguardo l'occupazione volumetrica del robot (C) e viene implementato l'algoritmo di ricerca di percorso (D). Il percorso è tracciato con un controllore industriale (E) e l'intero codice del progetto, più alcuni valori di riferimento per un'analisi comparativa, sono forniti per riferimenti futuri (F). Nella sezione V vi è una discussione generale che verte sulle limitazioni correnti del metodo introdotto da questa pubblicazione e del suo studio. Nella sezione VI le conclusioni vertono attorno ai vantaggi e svantaggi del suddetto metodo. Per concludere, nella sezione VII sono fornite osservazioni e deviazioni perfettamente ragionevoli dalle vie battute da esplorare nei lavori futuri.

## II. LETTERATURA SCIENTIFICA AGGIUNTIVA

Tipici algoritmi di pianificazione di percorso o di traiettoria in intelligenza artificiale sono  $A^*$  [21] e le sue varianti quali *improved  $A^*$*  {lett.  $A^*$  migliorato} [22], *double  $A^*$*  {lett. doppio  $A^*$ } [8], *hierarchical  $A^*$*  {lett.  $A^*$  gerarchico} [23], o l'algoritmo di Dijkstra e le sue varianti [24]. Questi sono solamente alcuni esempi, tra centinaia. Una sottoclasse di questi ed altri algoritmi sono specificamente progettati per i manipolatori industriali, come ad esempio il già citato *double  $A^*$*  {lett. doppio  $A^*$ } [8], il *non-probabilistic anytime algorithm* {lett. algoritmo non-probabilistico ad ogni tempo} [25], i *potential fields* {lett. campi di potenziale} [26,27], le *probabilistic roadmaps* {lett. carte stradali probabilistiche} [18,19] e i già citati RRT [17]. Kallman *et al.* [28] hanno proposto le *dynamic roadmaps* {lett. carte stradali dinamiche} con gli RRT bidirezionali per ovviare agli ambienti in costante cambiamento. Ci sono anche versioni ibride come il *cell-based Voronoi roadmap generation algorithm* {lett. algoritmo di generazione di carte stradali basato su celle di Voronoi} [10] che è esplorato con  $A^*$ . Sono stati fatti dei tentativi con il reinforcement learning {lett. apprendimento a rinforzo} [29,30]. Tutti questi algoritmi sono nel più dei casi basati su spazi di configurazione ad alta dimensionalità, corrispondenti alle posizioni dei giunti robotici. Alcuni sono basati su sistemi di coordinate cartesiani che corrispondono allo spazio di lavoro robotico in 2D o in 3D. Con rare eccezioni, come ad esempio un approccio a coordinate polari per la pianificazione di percorso di robot mobili utilizzando una griglia a ventaglio [31], nessun altro tipo di ricerca in spazi radiali sembra essere stata adottata per la modellazione dei movimenti dei robot industriali.

Llopis-Albert *et al.* [32] hanno presentato una comparazione dei più diffusi algoritmi per la pianificazione di traiettoria basati su ricerca. Gli algoritmi di ricerca basati su spazi multidimensionali sono maggiormente legati alla possibilità data da un certo tipo di modello robotico nel fare i calcoli principali nel suo spazio di configurazione, anche chiamato spazio dei giunti, con molti gradi di libertà ridondanti (DOF), per esempio sei o sette. Kaltsoukalas *et al.* [33] hanno proposto un algoritmo di ricerca che genera una

griglia di punti alternativi basati su configurazioni simili per il percorso che conduca alla posizione e all'orientamento desiderato per l'estremità (EE) del robot.

### III. SIMMETRIE CILINDRICHE DEI ROBOT INDUSTRIALI

I più comuni manipolatori robotici industriali sono manipolatori seriali [34], composti da una serie di collegamenti rigidi connessi da giunti rotativi o prismatici che per semplicità di controllo sono solamente ortogonali, paralleli o intersecanti gli assi dei giunti. Spesso la struttura richiama un braccio antropomorfo che può essere descritto da una spalla, un gomito ed un polso. Il numero convenzionale di giunti è sei perché ci vogliono almeno sei gradi di libertà (DOF) per muovere l'estremità del robot verso un'arbitraria posizione ed orientamento nello spazio. Quattro DOF sono tuttavia utilizzati per operazioni di manifattura più semplici.

Il lavoro che ha più influenzato la produzione delle comuni strutture di manipolatori robotici disponibili oggi è la struttura cinematica 321 di Donald L. Pieper [35] che fornisce una soluzione della cinematica inversa in forma chiusa per i manipolatori seriali con sei giunti rotativi e con tre giunti consecutivamente intersecanti. Le simmetrie prodotte da questa soluzione sono ulteriormente utilizzate in questo articolo scientifico, come mostrato in questa sezione, fino ad ottenere uno spazio discreto a configurazione cilindrica che non richiede calcoli della cinematica diretta o inversa per la pianificazione di percorso o di traiettoria.

#### A. SIMMETRIE CILINDRICHE DI UN ABB IRB 1600

Un robot industriale ABB IRB 1600 è utilizzato in questo articolo scientifico come esempio di un tipico robot industriale a sei DOF. Frequentemente si usano i primi tre giunti rotativi,  $J_1$ ,  $J_2$  and  $J_3$ , per muovere l'estremità (EE) nello spazio, e altri tre giunti rotatori,  $J_4$ ,  $J_5$  and  $J_6$ , in una configurazione a polso in modo da impostare accuratamente l'orientazione dell'EE. Si veda la figura 1 per riferimento.

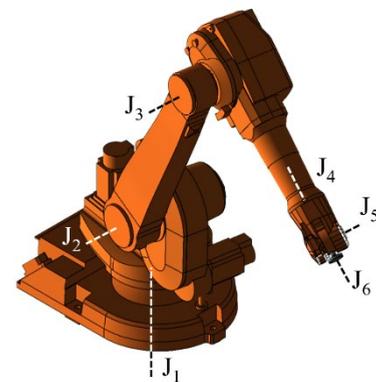
La simmetria cilindrica utilizzata da questo articolo scientifico riguarda tutti i manipolatori robotici con il primo giunto rotativo  $J_1$ . Per il robot ABB IRB 1600,  $J_1$  può ruotare di  $\pm 180$  gradi dalla posa di base. Si noti che quel che segue in questa pubblicazione non è applicabile se il primo giunto del robot utilizzato non è di tipo rotativo. Tuttavia, altre simmetrie potrebbero essere utilizzabili e questo articolo scientifico pone le basi per un più esteso approccio procedurale alla pianificazione di percorso o traiettoria in presenza di simmetrie.

Per lo stesso robot in analisi,  $J_2$  e  $J_3$  sono di tipo rotativo e allineati sullo stesso piano. In particolare,  $J_2$  e  $J_3$  funzionano congiuntamente come un gomito per il manipolatore robotico: una combinazione dei due valori angolari conduce a qualsiasi posizione raggiungibile in coordinate polari sul dato piano. Ogni posizione può essere raggiunta al più in due configurazioni, rispettivamente con il gomito piegato all'interno o all'esterno. Al fine di semplificare il caso d'uso

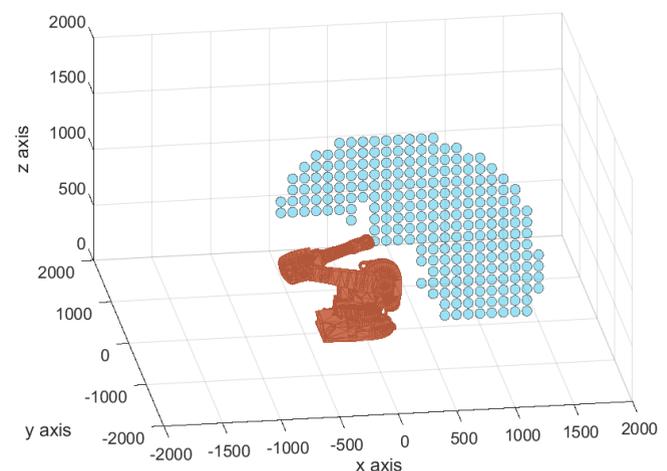
portato a dimostrazione di questo articolo scientifico, sono prese in considerazione solamente le configurazioni con gomito interno.

$J_4$ ,  $J_5$  e  $J_6$  sono maggiormente dedicati alla definizione dell'orientamento dell'EE, siccome possono solamente variare di poco la posizione dell'EE, al contrario di  $J_1$ ,  $J_2$  e  $J_3$  che hanno influenza maggiore su questa.

La simmetria cilindrica è prodotta bloccando l'EE del robot con un certo orientamento sul piano cartesiano  $xz$  con  $y = 0$  e muovendo discretamente la posizione dell'EE sullo stesso piano usando al massimo le variazioni di  $J_2$  e  $J_3$  con gomito in configurazione interna o esterna. Il risultato di questa operazione può essere chiamato piano di raggiungibilità per la data orientazione dell'EE. Tutti i valori dei giunti per ciascuna posizione dell'EE su ciascuno piano di raggiungibilità sono quindi calcolati *offline* {si legga "in anticipo"}. Si veda la sezione IV (A) per maggiori dettagli su come questi valori siano collezionati per la dimostrazione presentata in questa pubblicazione.



**FIGURA 1.** Vista 3D di un ABB IRB 1600 con giunti  $J_2$  e  $J_3$  che formano un gomito e giunti  $J_4$ ,  $J_5$  e  $J_6$  che formano un polso. Il giunto  $J_1$  ha un ruolo chiave nella produzione delle simmetrie dei robot industriali che sono utilizzate in questa ricerca. Le linee tratteggiate indicano gli assi di rotazione per ciascun giunto.



**FIGURA 2.** Vista 3D del piano di raggiungibilità  $xz$  (con  $y = 0$ ) del ABB IRB 1600 per gomito INWARD e orientamento del polso CENTER. L'intersezione tra il piano di base del robot e l'asse di rotazione del giunto rotativo  $J_1$  è l'origine del sistema di coordinate.

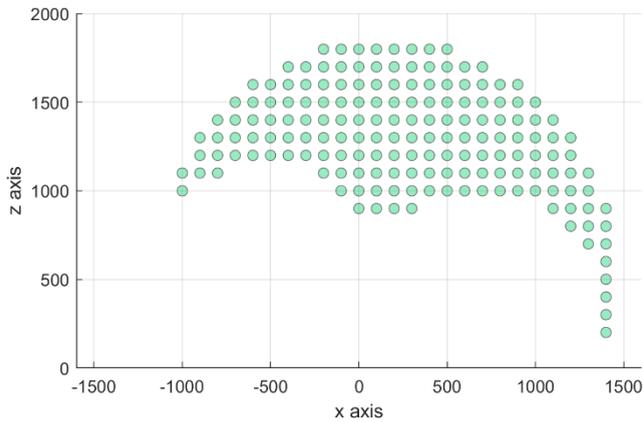


FIGURA 3. Vista in 2D del piano di raggiungibilità  $xz$  (con  $y = 0$ ) del robot ABB IRB 1600 per gomito INWARD e orientamento di polso UP.

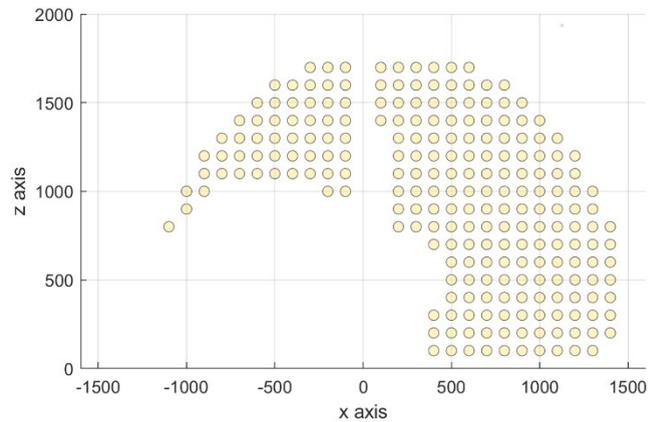


FIGURA 6. Vista in 2D del piano di raggiungibilità  $xz$  (con  $y = 0$ ) del robot ABB IRB 1600 per gomito INWARD e orientamento di polso LEFT.

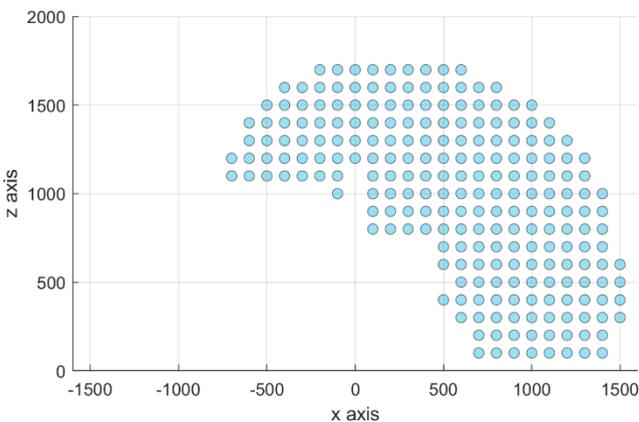


FIGURA 4. Vista in 2D del piano di raggiungibilità  $xz$  (con  $y = 0$ ) del robot ABB IRB 1600 per gomito INWARD e orientamento di polso CENTER.

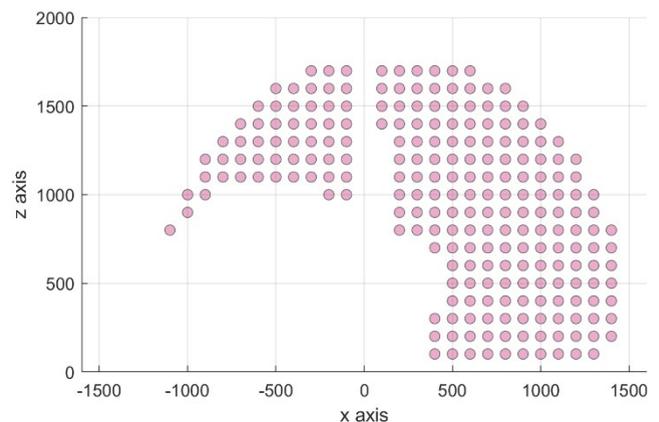


FIGURA 7. Vista in 2D del piano di raggiungibilità  $xz$  (con  $y = 0$ ) del robot ABB IRB 1600 per gomito INWARD e orientamento di polso RIGHT.

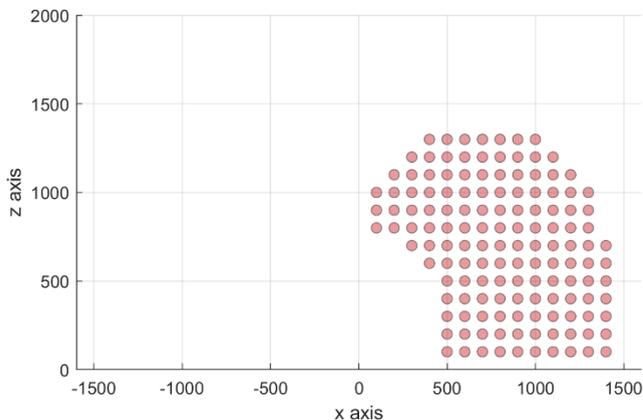


FIGURA 5. Vista in 2D del piano di raggiungibilità  $xz$  (con  $y = 0$ ) del robot ABB IRB 1600 per gomito INWARD e orientamento di polso DOWN.

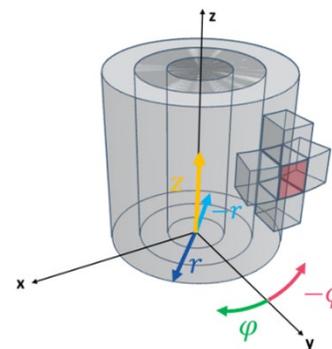
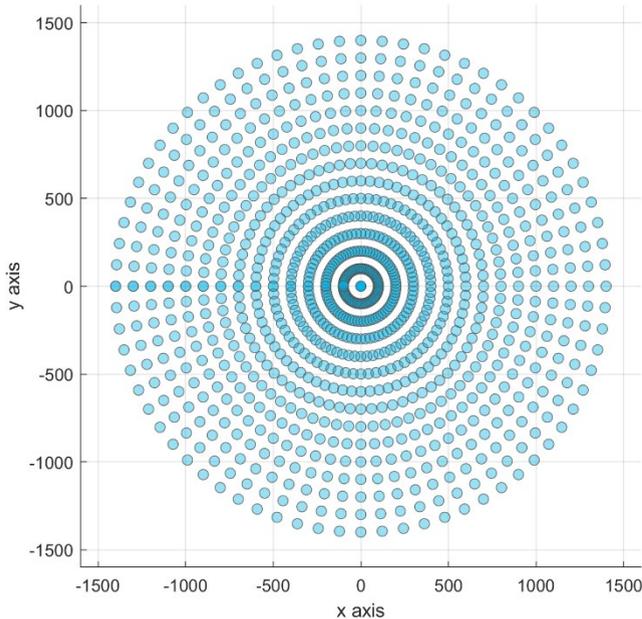


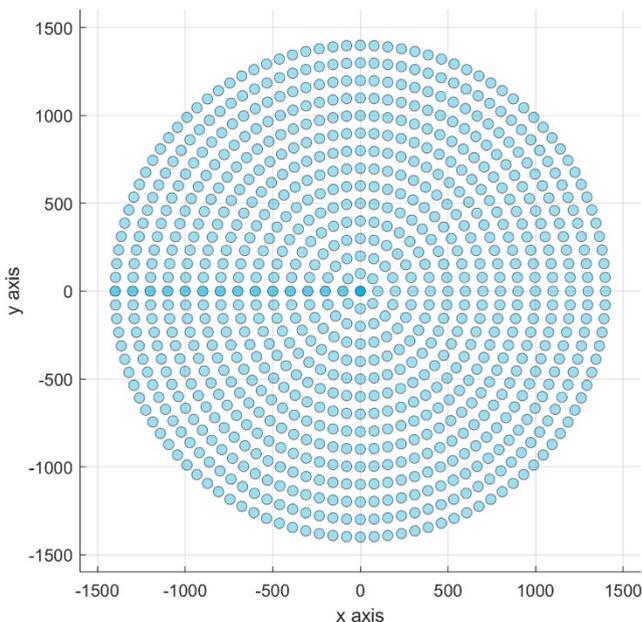
FIGURA 8. Sistema di coordinate cilindriche e alcune celle della griglia cilindrica. Una cella è identificata dalla sua posizione centrale in coordinate cilindriche.

TABELLA 1. Limiti dei giunti (angoli espressi in gradi) per i piani di raggiungibilità.

Piano di raggiungibilità	Limiti di $J_1$ (gradi)		Limiti di $J_2$ (gradi)		Limiti di $J_3$ (gradi)		Limiti di $J_4$ (gradi)		Limiti di $J_5$ (gradi)		Limiti di $J_6$ (gradi)	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
CENTER	0	0	-88.88	95.68	-84.63	65	0	0	-113.52	112.93	0	0
UP	0	0	-89.75	103.14	-84.44	59.81	0	0	-114.78	52.12	0	0
DOWN	0	0	-66.47	98.07	-76.23	64.10	0	0	-32.89	114.39	0	0
LEFT	-33.02	33.02	-88.75	94.54	-81.79	67.24	-122.88	-58.80	-113.91	-67.30	-137.44	158.45
RIGHT	-33.02	33.02	-88.75	94.54	-81.79	67.24	58.80	122.88	-113.91	-67.30	-158.45	137.44



**FIGURA 9.** Un piano cartesiano  $xy$  che mostra la griglia polare per ogni altezza  $z$  della griglia cilindrica. La densità della griglia polare è variabile perché vi è un numero costante di punti per raggio. La griglia si sovrappone per angoli  $\varphi = \pm\pi$ . Malgrado la densità non sia ideale, questa configurazione consente ai piani  $xz$  di ruotare attorno all'asse  $z$ .



**FIGURA 10.** Un piano cartesiano  $xy$  che mostra la griglia polare per ogni altezza  $z$  della griglia cilindrica. La densità della griglia polare è costante perché vi è un numero variabile di punti per raggio. La griglia si sovrappone per angoli  $\varphi = \pm\pi$ . Questa configurazione ha una densità ideale ma non consente ai piani  $xz$  di ruotare attorno all'asse  $z$  senza una perdita di dati.

Un database di conoscenza è creato e riempito con tutti i valori di mappaggio tra predefinite posizioni ed orientamenti dell'EE in coordinate cilindriche ed i rispettivi valori angolari dei giunti robotici. Questa operazione consente di sostituire i calcoli cinematici diretti e inversi necessari per la

pianificazione di percorso o di traiettoria con una ricerca in spazio a coordinate cilindriche discrete.

Alcuni piani di raggiungibilità per  $J_2$ - $J_3$ -interni o -esterni possono essere ottenuti bloccando  $J_1$ ,  $J_4$  e  $J_6$  al valore zero. Per diversi valori di  $J_2$ ,  $J_3$  e  $J_5$ , il piano  $xz$  con  $y = 0$  dello spazio di lavoro cartesiano possono essere chiamati CENTRAL {lett. centrale}, UP {lett. sopra} e DOWN {lett. sotto}, basandosi sulla configurazione data al polso. Si veda la sezione IV (A) per ulteriori dettagli su queste configurazioni di polso. Alternativamente, qualsiasi valore dei giunti può essere usato per caratterizzare tutte le posizioni del piano  $xz$  con  $y = 0$  in altre specifiche configurazioni del polso.

Questo è il caso dei piani di raggiungibilità con configurazioni dei polsi LEFT {lett. sinistra} e RIGHT {lett. destra}. Si veda la sezione IV (A) per ulteriori dettagli circa queste configurazioni di polso. Vi è sempre un vantaggio nei calcoli e in un minore consumo energetico del robot se alcuni dei valori dei giunti possono essere mantenuti costanti durante il movimento planare, proprio perché i relativi motori non devono muoversi. Il fatto che meno movimenti dei giunti siano necessari per muovere discretamente la posizione dell'EE su una griglia, dato un orientamento fisso dell'EE, è ulteriormente utilizzato in questo metodo in qualità di pre-ottimizzazione alla pianificazione del percorso o della traiettoria.

In termini di coordinate cilindriche, ciascun piano di raggiungibilità  $xz$  con  $y = 0$  può essere ruotato cilindricamente attorno all'asse  $z$  variando i valori di  $J_1$  di un certo *offset* {si legga "valore predeterminato"} nei limiti di raggiungibilità del robot. Quindi, ciascuna posizione dell'EE può essere espressa in coordinate cilindriche in 3D  $(\varphi, z_{cyl}, r)$  con  $\varphi = J_{1,xz} + J_{1,offset}$ ,  $z_{cyl} = z_{cart}$  e  $r = x$  derivate dalle coordinate cartesiane in 2D sul piano  $xz$  con  $y = 0$  e l'*offset*  $J_1$ . Le coordinate cilindriche discrete sono mostrate in figura 8. Va notato che ciascuna posizione cilindrica in 3D può essere raggiunta al più da due punti di ciascun piano di raggiungibilità  $xz$  con  $y = 0$  perché ruotano attorno all'asse  $z$  che passa per il loro centro. Quindi un raggio può essere anche negativo. Questo non è un problema perché l'algoritmo di ricerca del percorso si prende cura delle coordinate in 4D associate a ciascuno dei punti e le valuta nelle loro diverse orientazioni di riferimento. Si veda la sezione IV(B) per maggiori dettagli.

I valori angolari di massimo e minimo dei giunti influenzano direttamente i limiti dello spazio discreto. Ciò può essere osservato dai piani di raggiungibilità nelle figure 2-7.

#### IV. RICERCA DI PERCORSI CILINDRICI IN 4D

Il metodo presentato in questo articolo scientifico accoppia lo spazio cilindrico creato attorno alle simmetrie dei robot industriali con un algoritmo in intelligenza artificiale di ricerca di percorso. Questa sezione copre tutti i passi verso una implementazione di successo di un algoritmo di ricerca di percorso in spazio cilindrico in 4D. I passi affrontati da questa

sezione vanno dalla pratica acquisizione dei valori dei giunti del robot per il database di conoscenza (A), alla definizione della quarta dimensione (B), la definizione dell'occupazione volumetrica del robot per applicazioni d'evitamento degli ostacoli (C), la ricerca con  $A^*$  sulla griglia discreta (D), il tracciamento del percorso con il controllore robotico (E) e alcuni valori di riferimento per un'analisi comparativa (E).

#### A. ACQUISIZIONE DEI VALORI DEI GIUNTI ROBOTICI

I punti necessari per caratterizzare ciascun piano di raggiungibilità per una data configurazione del polso sono campionati direttamente da RobotStudio<sup>®</sup>, il *software* di simulazione sviluppato da ABB. Questo passaggio può essere simulato anche tramite altri *software* matematici, ad esempio Matlab<sup>®</sup>, usando i modelli cinematici dei robot. Utilizzare ABB RobotStudio<sup>®</sup> permette ai ricercatori di assicurare l'accuratezza del modello robotico presente nel *software* di proprietà e quindi di ottenere valori dei giunti accurati.

Gli autori hanno testato l'approccio su cinque piani di raggiungibilità in 2D con gomito J2-J3-interno e orientamenti di polso CENTRAL, UP, DOWN, LEFT e RIGHT. Ogni piano è caratterizzato da un numero finito di posizioni dell'EE che rientrino nei limiti di raggiungibilità del robot in quel particolare orientamento, si vedano le figure 3-7. Le posizioni sono distanziate da 100mm in modo da formare una griglia cartesiana discreta in 2D, una scelta che è puramente fatta dagli sviluppatori in dipendenza dai requisiti dell'applicazione, in questo caso semplicemente per testare alcune traiettorie nello spazio libere da ostacoli. Quindi, dopo aver campionato tutti i valori dei giunti per tutte le posizioni dell'EE su un dato piano di raggiungibilità, lo spazio cilindrico 3D risultante viene discretizzato.

La procedura ha prodotto 226 coordinate per il piano CENTRAL (si veda la figura 4), 169 coordinate per il piano UP (si veda la figura 3), 140 coordinate per il piano DOWN (si veda la figura 5), 228 coordinate per il piano LEFT (si veda la figura 6) e 228 coordinate per il piano RIGHT (si veda la figura 7). I limiti dei giunti sono riportati nella tabella 1.

La densità della griglia cilindrica dipende da una scelta che è specifica per la forma cilindrica: quanti valori discreti sono considerati per ogni valore del raggio. L'unica scelta che consente ai piani  $xz$  di essere rotati angolarmente attorno all'asse  $z$  in modo da evitare la perdita di punti in uno spazio cilindrico in 3D è mantenere il numero di coordinate per raggio costante. Questo produce una densità variabile nella griglia cilindrica in 3D (si veda la figura 9). Infatti, le coordinate ai raggi vicini all'asse  $z$  sono più dense di quelle ai raggi lontani dall'asse  $z$ . Il criterio che consente di scegliere un adeguato numero di punti per raggio è di valutare la distanza di due coordinate contigue al raggio maggiore ed approssimarla alla distanza cartesiana usata per le coordinate planari, così che l'angolo risultante divida una circonferenza in  $\varphi - 1$  settori e produca  $\varphi$  posizioni dei piani di coordinate angolari. Il risultato di questo calcolo per il robot ABB IRB

1600 con una distanza cartesiana di 100mm produce un valore ottimale di 72 punti e/o piani per raggio o 72 coordinate angolari  $\varphi$  all'estremo raggio di 1500mm dall'asse  $z$ . Dal punto di vista dell'algoritmo di ricerca, spostarsi in una griglia cartesiana o una griglia cilindrica con un numero costante di punti per raggio (densità variabile) è la stessa cosa. Ciò è anche il motivo per cui questa scelta è un buon compromesso.

La scelta di ottimizzare il numero di punti per raggio ha il vantaggio di mantenere la griglia cilindrica il più costante possibile. Questo però distruggerebbe inevitabilmente la continuità dei piani radiali. Pertanto, si renderebbe l'operazione di ricerca più ardua, con un sistema di coordinate incoerente; ad esempio vi sarebbero  $\varphi$  salti da un raggio all'altro, come mostrato in figura 10. Questa alternativa è stata utilizzata almeno in un altro articolo scientifico che presenta un approccio alla pianificazione di percorso per robot mobili con sistema di coordinate polari a forma di ventaglio [31]: gli autori hanno optato per una densità variabile in questo lavoro.

72 punti per raggio corrispondono a rotazioni angolari di 5 gradi per ciascun piano di raggiungibilità. Perciò, 72 volte i punti originari sono automaticamente prodotti ruotando solamente il giunto  $J_1$  per valori multipli di 5 gradi per tutte le coordinate che riguardano la rotazione di un piano di raggiungibilità. Lo spazio cilindrico in 3D discreto conta 16272 coordinate per il piano CENTRAL, 12168 coordinate per il piano UP, 10080 coordinate per il piano DOWN, 16416 coordinate per il piano LEFT e 16416 coordinate per il piano RIGHT. Per un totale di 71352 coordinate cilindriche in 3D, con valori angolari dei giunti  $J_1, J_2, J_3, J_4, J_5$  e  $J_6$  conservati nel database per ogni coordinata.

Lavorare nello spazio cilindrico è preferibile al lavorare nello spazio delle configurazioni perché tre dimensioni possono ancora essere visualizzate nello stesso grafico e la trasformazione da coordinate cartesiane a cilindriche è molto più semplice dell'applicare le equazioni della cinematica diretta o inversa.

Bisogna fare una nota sui piani di raggiungibilità LEFT e RIGHT:  $J_1$  è già ruotato di una quantità variabile lungo il piano  $xz$  con  $y = 0$  per poter compensare la rotazione angolare dell'EE. Questa quantità ha un valore massimo di  $33.22^\circ$  lungo il piano che deve essere rimossa dalla rotazione massima di  $J_1$  attorno all'asse  $z$ . Questo spezza la griglia cilindrica per  $J_1 = \pm 146.78^\circ$ , che sarebbe altrimenti chiusa su  $J_1 = \pm 180^\circ$  con coordinate sovrapposte.

#### B. QUARTA DIMENSIONE

Tre dimensioni non sono abbastanza e sei sono troppe. Per poter navigare in ciascuno spazio cilindrico risultate da ciascun orientamento di polso, una quarta dimensione chiamata "orientamento" (abbr. "o") è aggiunta allo spazio cilindrico discreto in 3D. A ciascuno spazio cilindrico viene quindi data una coordinata numerica discreta, a scelta degli autori, con valori da 1 a 5, rispettivamente corrispondenti agli orientamenti di polso CENTRAL, UP, DOWN, LEFT e

RIGHT. Lo spazio cilindrico discreto in 4D si esprime con coordinate discrete  $(o, \varphi, z, r)$  nelle quali  $o$  è un numero convenzionale assegnato a ciascun orientamento di polso (ad esempio l'orientamento di polso CENTRAL corrisponde a  $o = 1$ ),  $\varphi$  rappresenta il numero di spostamenti angolari di 5 gradi positivi o negativi dall'asse  $y$  (si veda la figura 8),  $z$  corrisponde alla stessa altezza in coordinate cartesiane discretizzata nel numero di unità di distanza (in questo caso 100mm) dall'origine, ed  $r$  corrisponde al raggio discreto, positivo o negativo, discretizzato nel numero di unità di distanza (in questo caso 100mm) dall'origine (si vedano le figure 8 e 9). Il passaggio da coordinate continue cartesiane in 3D a coordinate continue cilindriche in 3D e *vice versa* è una semplice operazione matematica che fa uso di seno e coseno. Dalle coordinate cartesiane alle coordinate cilindriche si usi:

$$\begin{cases} r^2 = x^2 + y^2 \\ \tan \varphi = \frac{y}{x} \\ z = z \end{cases} \quad (1)$$

Si noti che la funzione inversa della tangente ha un numero di soluzioni infinito ma che può essere ristretto ad un'unica soluzione se il quadrante dove è localizzato il punto è conosciuto. Queste regole sono quasi sempre presentate nei libri di matematica pertanto si rimanda la ricerca al lettore. Da coordinate cilindriche a cartesiane la trasformazione è più semplice perché seno e coseno danno soluzioni uniche:

$$\begin{cases} x = r \cos \varphi \\ y = r \sin \varphi \\ z = z \end{cases} \quad (2)$$

Gli unici ulteriori requisiti sono la discretizzazione delle coordinate cilindriche ottenute, per la quale si usano i valori discreti più prossimi disponibili sulla griglia così come definita per l'applicazione, e l'orientamento di polso che si specifica come quarta coordinata cilindrica, in accordo con i valori definiti per convenzione nell'applicazione data. L'intervallo di valori della griglia influenza direttamente il tempo computazionale (più lungo per una griglia fine). Ciò significa che va fatto un compromesso per soddisfare i bisogni operativi e mantenere il tempo necessario per la computazione ragionevolmente breve. Questo nuovo sistema di coordinate consente ad un algoritmo di ricerca di risolvere automaticamente il problema di ricerca di percorso o di traiettoria partendo da una specifica configurazione ed arrivando ad un'altra specifica configurazione. Tutte le configurazioni di transito o esistono nello spazio di ricerca o non sono possibili. Questo risolve automaticamente anche il problema dei possibili punti di singolarità perché questi non sono salvati nello spazio di ricerca, cioè tra le coordinate robotiche discrete di ogni piano di raggiungibilità, senza il costo d'uso dell'intero spazio delle configurazioni del robot in sei o più dimensioni. La supposizione fatta nello spazio discreto è che tutti i movimenti dei giunti tra celle adiacenti siano continui. Questo è possibile grazie alla piccola distanza

presente tra celle che consente al robot di mantenere la stessa configurazione mentre si muove l'EE tra cella e cella, e oltretutto non si incontrano punti di singolarità.

### C. OCCUPAZIONE VOLUMETRICA DEL ROBOT

Il volume occupato dal corpo del robot per ciascuna delle coordinate in 4D può essere interamente calcolato *offline* con un modello di disegno assistito dal calcolatore (CAD) del robot o con alcuni involucri di contenimento attorno alle sue parti. Questi possono essere salvati nel database di conoscenza per un recupero *online* {si legga "durante l'esecuzione"}. Ciò significa che se degli ostacoli statici sono presenti nello spazio di lavoro del robot, tutte le celle che ne sono condizionate possono essere rimosse dallo spazio di ricerca. Non solamente le celle che intersecano direttamente gli ostacoli, ma anche tutte quelle per le quali il corpo robot è in collisione con gli ostacoli.

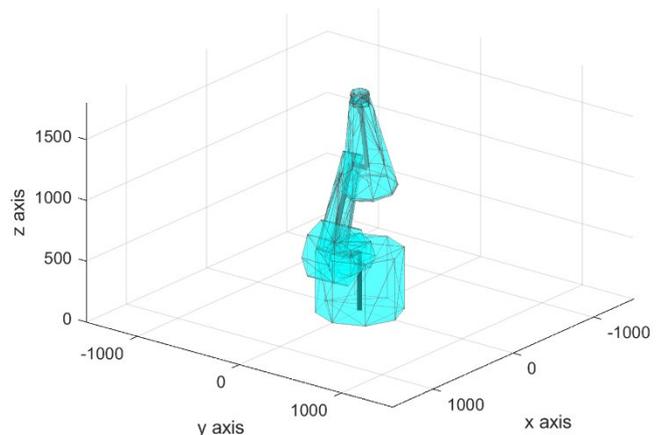


FIGURA 11. Lo scheletro che traccia la notazione DH per il robot ABB IRB 1600 in nero e gli involucri di contenimento a coni troncati che rappresentano l'occupazione volumetrica del robot industriale.

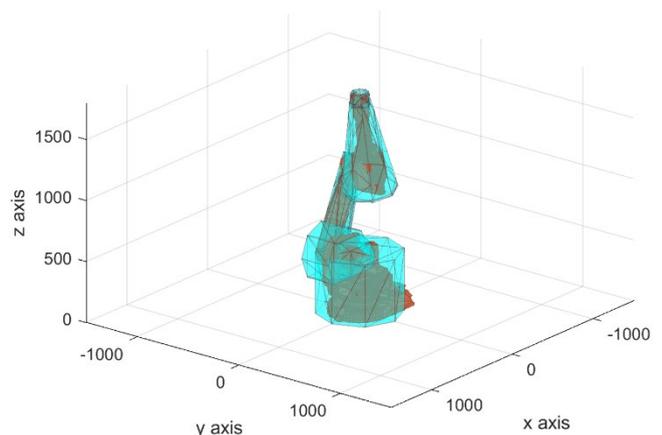


FIGURA 12. Involucri di contenimento a coni troncati che racchiudono le parti mobili del robot industriale ABB IRB 1600.

La qualità degli involucri di contenimento dipende da un compromesso tra il tempo computazionale e l'accuratezza necessaria durante la scansione *offline* dello spazio di lavoro del robot. L'evitamento degli ostacoli statici non richiede una

veloce riscansione della scena quanto l'evitamento degli ostacoli dinamici (in movimento). La scelta è ovviamente dipendente dall'applicazione robotica industriale. Per la dimostrazione in questo articolo scientifico vengono considerati solamente ostacoli statici e gli involucri di contenimento sono coni troncati centrati su uno scheletro del robot prodotto dal tracciamento lineare dei punti cardinali della notazione di Denavit e Hartenberg (DH) per il robot ABB IRB 1600 robot (si vedano le figure 11-12).

#### D. ALGORITMO DI RICERCA DI PERCORSO A\*

L'obiettivo di questo lavoro è di muovere il robot con una ricerca di percorso discreto su una griglia cilindrica in 4D, che è una griglia cilindrica 3D maggiorata da una quarta coordinata relativa all'orientamento discreto del polso del robot. La ricerca di un percorso ottimale e libero da ostacoli in uno spazio cilindrico a 4D è effettuata usando, tra i vari possibili algoritmi presentati nell'introduzione di questo articolo scientifico, un comune algoritmo di ricerca in intelligenza artificiale chiamato A\* {si legga "A star"}. Resta inteso che la novità introdotta da questa sezione riguarda il mostrare come uno di questi algoritmi, più o meno efficace in dipendenza dei requisiti applicativi, possa essere usato per la ricerca di percorso sulla griglia cilindrica in 4D proposta da questo articolo scientifico. In questo caso, l'obiettivo dell'algoritmo A\* è di trovare il percorso più breve tra due nodi di un dato spazio – un nodo di partenza ed un nodo d'arrivo – fintanto che vi sia un percorso. In termini tecnici, questo algoritmo è sia ottimo che completo. Ottimo perché il percorso è il più breve possibile, e completo perché se una soluzione esiste, è garantito che l'algoritmo la trovi.

La decisione ottimale dell'algoritmo A\* è basata sulla minimizzazione di una funzione di costo che è divisa in due componenti per ogni nodo  $n$ :  $h(n)$  e  $g(n)$ . Il valore  $h(n) = d(n_s, n)$  è l'esatto costo del percorso dal nodo di partenza  $n_s$  al nodo  $n$ , usando l'equazione (3) per calcolarlo, perché il percorso va attraversato nodo per nodo finché  $n$  non è raggiunto. L'altro valore  $g(n) = d(n, n_t)$  è un costo stimato dal nodo  $n$  al nodo d'arrivo  $n_t$ , usando la stessa equazione (3) per calcolarlo, perché il resto del percorso non è stato ancora trovato ed attraversato. Entrambi i valori sono calcolati usando la stessa funzione di costo (3) che in mappe bi- o tri-dimensionali è di solito una funzione semplice come la distanza euclidea tra due nodi. Nella mappa cilindrica a quattro dimensioni introdotta da questo articolo scientifico, la funzione di costo proposta e testata è una pseudo-distanza tra due nodi  $n_1(o_1, \varphi_1, z_1, r_1)$  e  $n_2(o_2, \varphi_2, z_2, r_2)$ :

$$d(n_1, n_2) = w(o_1, o_2) + \sqrt{(r_2 - r_1)^2 + \left(\frac{2\pi R}{n_\varphi}\right)^2 (\varphi_2 - \varphi_1)^2 + (z_2 - z_1)^2} \quad (3)$$

dove

$$w(o_1, o_2) = \begin{cases} 1 & \text{if } o_1 = o_2 \\ 0 & \text{altrimenti} \end{cases} \quad (4)$$

è la pseudo-distanza tra due coordinate d'orientamento,

$$R = \begin{cases} 0 & \text{if } r_1 \geq 0 \text{ e } r_2 \leq 0 \\ 0 & \text{if } r_1 \leq 0 \text{ e } r_2 > 0 \\ \min(|r_1|, |r_2|) & \text{altrimenti} \end{cases} \quad (5)$$

è il raggio per il quale lo spostamento tra  $\varphi_1$  e  $\varphi_2$  è calcolato, e  $n_\varphi$  è la densità angolare  $\varphi$  al raggio  $R \neq 0$ .

Si noti che  $r_1$  e  $r_2$  possono essere sia positivi che negativi perché i piani di raggiungibilità  $xz$  con  $y = 0$  sono ruotati attorno all'asse  $z$  attorno alla coordinata  $x = 0$ .

L'algoritmo di ricerca crea due liste di nodi: aperti e chiusi. All'inizializzazione, il nodo di partenza è aggiunto alla lista aperta e tutti i nodi nello spazio di ricerca che sono condizionati da ostacoli sono aggiunti alla lista chiusa. Un numero massimo di interazioni  $N$  è definito per evitare una ricerca infinita.

I passi seguenti sono ripetuti finché un percorso non viene trovato:

- 1) Se il numero massimo di interazioni  $N$  non è raggiunto, il nodo  $n$  nella lista aperta con il costo minore

$$f(n) = h(n) + g(n) \quad (6)$$

è selezionato come nodo corrente.

- 2) Il nodo corrente  $n$  viene spostato nella lista chiusa e si controlla se trattasi del nodo d'arrivo. Nel qual caso, la ricerca viene interrotta perché un percorso è stato trovato.
- 3) Tutti i nodi adiacenti ad  $n$  sono considerati come possibili nodi successivi da essere inseriti nella lista aperta. I nodi che sono nella lista chiusa sono ignorati in modo da non imbattersi in ostacoli o muoversi in circolo.
- 4) Se i possibili nodi successivi non sono presenti nella lista aperta, allora vi sono inseriti e il loro costo è calcolato. Se sono già nella lista aperta, si controlla se il percorso corrente è più breve del percorso per il quale erano stati aggiunti alla lista aperta precedentemente. Per ogni caso affermativo, il costo del nodo è ricalcolato e il nodo predecessore ridefinito come  $n$ .

Quando il nodo d'arrivo è raggiunto, il percorso totale è calcolato a ritroso dal nodo d'arrivo al nodo di partenza, richiamando ogni nodo predecessore dalla lista chiusa.

#### E. TRACCIAMENTO DEL PERCORSO CON IL CONTROLLORE ROBOTICO

La posizione dei giunti, salvata come dati per ogni cella della griglia discreta, costituisce l'informazione principale necessaria al controllore robotico per tracciare il percorso trovato dall'algoritmo di ricerca. Eventualmente, le transizioni da cella a cella possono essere marcate con una specifica velocità massima o questa può essere calcolata basandosi sui requisiti applicativi per uno specifico robot industriale. Ad

ogni modo, qualsiasi controllore robotico generalmente muove il robot da una posizione, espressa in coordinate dei giunti, ad un'altra posizione con un comando a velocità costante. Questo utilizza velocità trapezoidali e tende a mantenere la velocità massima tra spostamenti continui dei giunti. Per il controllore dell'ABB IRB 1600, questa operazione è effettuata tramite uno specifico comando MOVEJ (muovi i giunti) in linguaggio RAPID. Nella dimostrazione implementata per questo articolo scientifico la velocità è mantenuta costante. Le istruzioni sono inviate da un ambiente di simulazione, ad esempio Matlab®, al controllore robotico tramite protocollo TCP/IP ed eseguite in tempo reale. ABB RobotStudio® consente all'utente di visualizzare i movimenti del robot in un ambiente di simulazione invece che inviando i comandi direttamente al robot. Questo facilita il test e il rilascio di codice senza il rischio di danneggiare un robot industriale. Il codice usato per il controllore del ABB IRB 1600 in RobotStudio® può essere trovato sul sito indicato alla sezione IV (F).

#### F. CODICE E VALORI DI RIFERIMENTO PER ANALISI COMPARATIVE

Oltre ai limiti dei giunti riportati nella tabella 1 e i cinque piani di raggiungibilità (si vedano le figure 3-7), alcune traiettorie di prova sono eseguite con A\* e qui riportate allo scopo di fornire dei valori di riferimento per analisi comparative di implementazioni future di questo algoritmo e sue varianti. Nella figura 13 è possibile vedere l'intera traiettoria tracciata dal robot ABB IRB 1600 in Matlab®. Questa parte dalle coordinate  $(o, \varphi, z, r) = (1, 37, 13, 24)$  corrispondenti all'orientamento di polso CENTER identificabile con il colore blu della traiettoria (in figura 13) e del volume (in figura 14). La traiettoria continua verso le coordinate  $(o, \varphi, z, r) = (1, 33, 13, 24)$  mantenendo l'orientamento CENTER. Poi procede all'insù verso le coordinate  $(o, \varphi, z, r) = (2, 33, 15, 24)$  cambiando anche l'orientamento in UP e il colore della traiettoria e del volume in verde. Successivamente svolta verso sinistra fino alle coordinate  $(o, \varphi, z, r) = (4, 28, 15, 24)$  cambiando ancora l'orientamento in LEFT e il colore della traiettoria e del volume in giallo. Poi si muove in basso verso le coordinate  $(o, \varphi, z, r) = (3, 28, 10, 24)$  cambiando l'orientamento in DOWN e il colore della traiettoria e del volume in rosso. In seguito, continua verso l'interno e a destra fino a raggiungere le coordinate  $(o, \varphi, z, r) = (5, 37, 10, 20)$  e cambiando l'orientamento in RIGHT e il colore della traiettoria e del volume in fucsia. Per finire, si muove esternamente verso le coordinate  $(o, \varphi, z, r) = (1, 37, 10, 24)$  cambiando l'orientamento in CENTER e il colore della traiettoria e del volume in blu. Ciascuno di questi cambiamenti di coordinate è pianificato ed eseguito utilizzando A\*.

Il codice usato in questo progetto è disponibile per il download dalla pagina GitHub pubblica al seguente indirizzo web: <https://github.com/andreadegiorgio/cylindrical-astar>.

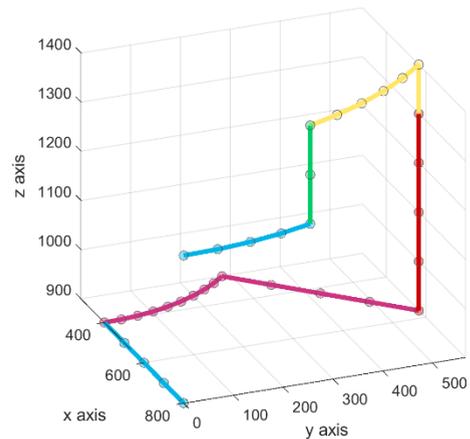


FIGURA 13. Traiettoria test per l'EEE visualizzata in Matlab®. Il robot si muove attraverso la griglia cilindrica in accordo al piano illustrato alla sezione IV (F). I colori indicano l'orientamento di polso CENTER (blu), UP (verde), DOWN (rosso), LEFT (fucsia) e RIGHT (giallo).

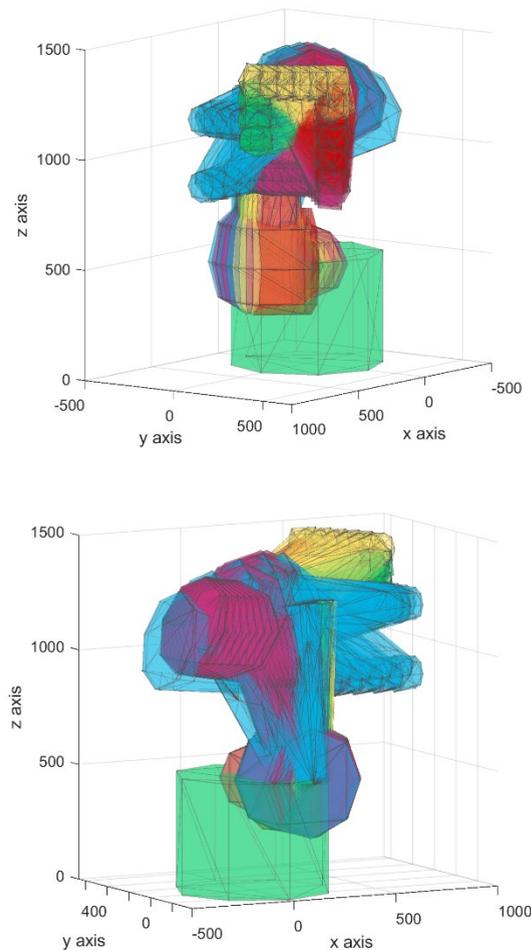
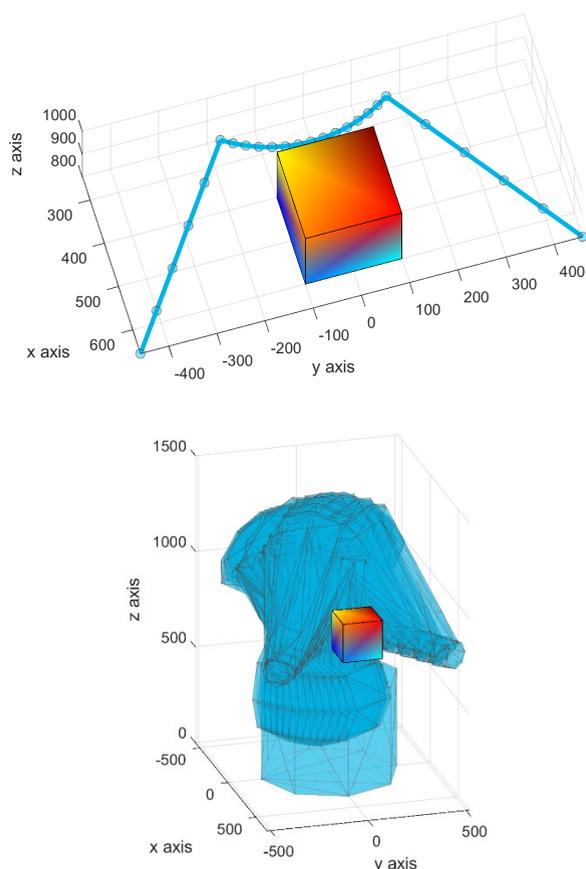


FIGURA 14. Occupazione di volume per involucri di contenimento a cono troncati del robot industriale ABB IRB 1600, sulla traiettoria di test della sezione IV (F) visualizzata in Matlab®. I colori indicano l'orientamento di polso CENTER (blu), UP (verde), DOWN (rosso), LEFT (fucsia) e RIGHT (giallo). Il volume è ruotato attorno all'asse z e mostrato da due diverse prospettive per motivi di visualizzazione.

Una singola traiettoria punto-punto addizionale viene presentata al fine di mostrare l'abilità di A\* di evitare gli ostacoli statici nello spazio in 4D. L'ostacolo definito per questo scopo è un cubo (si veda la figura 15) che rende impossibile per il robot muoversi direttamente da coordinate  $(o, \varphi, z, r) = (1,30,9,24)$  a coordinate  $(o, \varphi, z, r) = (1,44,9,24)$  approssimativamente in linea retta (si veda la figura 15, in alto). La configurazione del robot scelta è quella con orientamento di polso CENTER che appare nella simulazione Matlab® con il colore blu sia della traiettoria (si veda la figura 15, in alto) che del volume (si veda la figura 15, in basso). Da questo esempio si evince come A\* computi la più prossima alternativa tra le traiettorie che evitano collisioni tra il volume del robot industriale e il cubo. La simulazione della collisione è interamente effettuata in modalità *offline* e richiede circa 7 ore perché tutte le potenziali collisioni volumetriche del robot con gli ostacoli statici predefiniti siano verificate. La computazione di qualsiasi possibile traiettoria che eviti collisioni con gli ostacoli statici è fatta in modalità online e richiede secondi (fino ad alcuni minuti, in dipendenza della distanza tra punti d'arrivo e destinazione) per trovare un attendibile percorso libero da collisioni.



**FIGURA 15.** Traiettoria che evita un ostacolo (in alto nella figura). L'occupazione volumetrica del robot (in basso nella figura) mostra come il braccio robotico non tocchi l'ostacolo durante il tracciamento della traiettoria predefinita.

## V. DISCUSSIONE E LIMITAZIONI CORRENTI

La caratterizzazione dei robot industriali tramite le loro simmetrie, insieme con la necessità di modellare le configurazioni robotiche, ha portato alla creazione di un inesplorato spazio cilindrico in 4D. Tale spazio racchiude il potenziale per l'uso di vecchi e nuovi algoritmi di intelligenza artificiale in applicazioni industriali. L'intera *performance* dell'algoritmo di ricerca applicato è legata a diverse proprietà tra differenti modelli di robot, e a compromessi, inclusa la definizione della dimensione della griglia discreta. Per tale scelta dimensionale, ad esempio, altri lavori che vertono su ricerca in spazi discreti hanno presentato metodi che implementano ottimizzazioni a passo temporale e ulteriori miglioramenti della traiettoria apportati solamente quando vi sia tempo addizionale [36].

Risulta difficile paragonare il metodo presentato con algoritmi di ricerca di percorso o traiettoria *online* e *offline* [37], proprio perché questo cade a metà, con una ricerca *online* di uno spazio mappato *offline*. Per questa ragione, e per la mancanza di valori di riferimento per un'analisi comparativa, la *performance* dell'algoritmo di ricerca implementato in questo articolo scientifico non viene comparata con simili algoritmi di ricerca, e/o con metodi complementari o alternativi. Ai fini di fornire valori di riferimento per analisi comparative con implementazioni future di questo metodo di controllo robotico, alcuni valori della versione testata in questo articolo scientifico su un ABB IRB 1600 sono resi disponibili.

Una limitazione dell'approccio presentato in questo articolo scientifico è che l'algoritmo fa uso di informazione discreta circa le posizioni dei giunti robotici, che porta la ricerca indietro alle origini degli algoritmi di pianificazione di percorso o di traiettoria [38,39]. Ad ogni modo, gli algoritmi per pianificare movimenti robotici sono stati presentati primamente nei domini discreti, perché i computer potessero effettuare le complesse computazioni numeriche, e poi implementati nei domini continui, grazie agli avanzamenti nella potenza computazionale e nel *software* matematico. La ricerca futura deve indirizzare una definizione a grana fina, quasi continua, dello spazio discreto, così da rimuovere questa limitazione.

Una distinzione deve essere fatta tra pianificazione di percorso e traiettoria. Un percorso riguarda solo la definizione di un percorso spaziale dal punto A al punto B, senza nessuna informazione dinamica sul movimento lungo il percorso, come tempo, velocità, accelerazioni, etc. Quest'ultimo è infatti definito come traiettoria. In questo articolo scientifico sono trattate la pianificazione di percorso e solo parzialmente la pianificazione di traiettoria con velocità trapezoidali, legate all'introduzione dell'informazione su velocità o accelerazione nelle proprietà delle celle della griglia. L'adattamento di soluzioni esistenti più efficienti a questo nuovo approccio può essere considerato una direzione di ricerca futura.

L'evitamento di ostacoli statici e l'evitamento di ostacoli dinamici sono limitatamente suggeriti come un esempio di una possibile applicazione del approccio presentato e non sono estensivamente discussi in questo articolo scientifico. Mentre gli ostacoli statici possono essere immediatamente definiti come parte della mappa ricercata da un tradizionale algoritmo A\* (si veda la figura 15), gli ostacoli dinamici richiedono di usare algoritmi più avanzati quali ad esempio D\* (Dynamic A\* {lett. A\* dinamico}) [40], che non sono introdotti in questo articolo scientifico. D'altra parte, la definizione di occupazione volumetrica che viene fuori con il nuovo metodo può essere considerata un passo avanti sulle tecniche degli involucri di contenimento e la sua integrazione con gli algoritmi di evitamento degli ostacoli statici o dinamici può dirsi innovativa. La versione a coni troncati degli involucri di contenimento presentata in questo articolo scientifico è anch'essa una novità ed un compromesso tra modelli di robot accurati e involucri di contenimento tradizionali [41].

## VI. CONCLUSIONI

L'obiettivo di questo articolo scientifico è di caratterizzare le tipiche simmetrie di un manipolatore robotico industriale ed utilizzarle per applicare meglio gli algoritmi di controllo robotico ad intelligenza artificiale. Uno spazio a griglia cilindrica discreta in quattro dimensioni è il risultato di questo miglioramento e viene mostrato come questo possa direttamente sostituire modelli robotici complessi ai tempi di pianificazione ed esecuzione della traiettoria. L'applicazione di un algoritmo di intelligenza artificiale per la pianificazione di percorso, quale A\* che viene riproposto in questo articolo scientifico, sulla griglia cilindrica diviene una procedura innovativa che porta vantaggi e svantaggi.

**Meno equazioni.** Tra gli effetti desiderati è possibile condurre un robot industriale senza alcuna specifica conoscenza dei modelli cinematici e dinamici del robot al tempo di pianificazione e tracciamento di percorso o di traiettoria. L'informazione aggiuntiva può essere direttamente conservata nel database di conoscenza per ogni cella della griglia discreta, così come per l'occupazione volumetrica del corpo del robot ai fini dell'evitamento degli ostacoli e l'ottimizzazione di traiettoria alle massime velocità.

**Uno spazio di ricerca ridotto.** Lo spazio cilindrico in quattro dimensioni mostra entrambi i vantaggi dello spazio di configurazione e dello spazio di lavoro in 3D di un robot industriale. Ciò migliora lo stato dell'arte. Infatti, lo spazio di ricerca in 4D è un miglioramento rispetto all'intero spazio di configurazione esplorato da altri algoritmi perché trattiene la semplicità di ricercare in uno spazio in 3D, cilindrico invece che cartesiano, mentre mantiene precisa conoscenza della configurazione robotica sintetizzata in una sola coordinata aggiuntiva.

Il compromesso principale è tra una limitata capacità di computare informazioni di griglia *offline* e la velocità di

ricerca del percorso. Questo studio incoraggia l'uso degli algoritmi di ricerca per applicazioni di robotica industriale, e richiama alla ricerca di ulteriori simmetrie da sfruttare che siano presenti nei diversi robot industriali.

**Nessuna singolarità.** Tra i vantaggi dello spazio in 4D, i punti di singolarità vengono automaticamente esclusi dallo spazio quando i valori di posizione sono collezionati. L'abilità di saltare il controllo delle singolarità nella generazione di traiettorie costituisce una grande semplificazione di un modello robotico industriale.

**Nuovi algoritmi per l'evitamento di collisioni.** Gli algoritmi d'evitamento di ostacoli dinamici possono far uso del presente studio come base per nuove implementazioni. Uno svantaggio che può essere delineato è l'insufficiente capacità computazionale per velocizzare le ricerche negli spazi più ricchi di dati; quelli in cui, ad esempio, la grana della griglia diventa più fine, oppure dove vengono aggiunte alle celle informazioni sempre più complesse, quali l'occupazione volumetrica o altre. Fortunatamente si è dimostrato come la capacità computazionale non sia un ostacolo se non per alcuni anni, quando gli avanzamenti tecnologici la raddoppiano abbastanza in fretta.

**Potenziale per la collaborazione uomo-robot.** Le traiettorie basate sulla griglia cilindrica sono più comprensibili dagli operatori umani che lavorano con i robot industriali. Questo perché tali traiettorie sono centrate sul robot e orientate verso l'uomo, se si considera l'orientamento di polso. Al contrario, le traiettorie basate sugli spazi cartesiani o di configurazione sono meno rappresentative degli ambienti di collaborazione uomo-robot. Infatti, queste sono principalmente allineate con gli oggetti nello spazio di lavoro oppure non allineate affatto con nulla che possa essere usato come riferimento per comprendere le intenzioni del robot e i suoi movimenti.

## VII. LAVORO FUTURO

L'algoritmo presentato in questo articolo scientifico apre la via per ulteriori direzioni di ricerca ed altre deviazioni perfettamente ragionevoli dalle vie battute. Alcune di esse sono delineate qui di seguito con alcune ulteriori osservazioni dal presente studio.

**Montaggio degli attrezzi robotici.** L'operazione di montaggio di un attrezzo sopra l'EE del robot e di considerare il punto centrale dello strumento (TCP) come punto di riferimento per le coordinate dei giunti immagazzinate nel database di conoscenza invalida tutti i valori collezionati in assenza dello strumento. Quindi, l'introduzione di ogni nuovo strumento richiede una quantità di lavoro equivalente per produrre nuovi dati *offline* per l'algoritmo di ricerca.

**Orientamenti di polso aggiuntivi.** L'introduzione di nuovi orientamenti di polso, ad esempio UP-LEFT {lett. sopra-sinistra} or UP-RIGHT {lett. sopra-destra}, non richiede di cambiare il numero di dimensioni dello spazio di ricerca, bensì il numero di coordinate di orientamento. Infatti, ogni nuova

configurazione robotica corrisponde alla definizione di un nuovo piano di raggiungibilità  $xz$  con  $y = 0$  e una nuova coordinata di orientamento a cui riferirlo.

**Configurazioni di gomito addizionali.** L'introduzione della configurazione di gomito esterna, omessa in questo articolo scientifico, non richiede necessariamente un cambiamento nel numero delle dimensioni dello spazio di ricerca, ma semplicemente del numero di coordinate di orientamento. Infatti, ogni nuova configurazione robotica corrisponde alla definizione di un nuovo piano di raggiungibilità  $xz$  con  $y = 0$  e una nuova coordinata d'orientamento a cui riferirlo. Tuttavia, in questo caso ci dovranno essere delle regole su come transitare da coordinate con un orientamento di gomito intero a coordinate con un orientamento di gomito esterno, perché il braccio può muoversi da una configurazione robotica ad un'altra solamente per alcune specifiche coordinate della griglia, quando la distanza tra i valori dei giunti è minima.

**Gradi di libertà (DOF) robotici aggiuntivi.** La dimensionalità dello spazio di ricerca non dipende dalla dimensionalità dello spazio di configurazione. Ogni DOF addizionale per il robot industriale è compatibile con l'algoritmo di ricerca sulla griglia cilindrica discreta.

**Altri modelli di robot industriali.** L'abilità di sfruttare le simmetrie che sono specifiche dei modelli di robot industriali rende questo algoritmo altamente dipendente dai modelli di robot stessi. Fortunatamente, l'industria si è assestata su una quantità finita di modelli robotici che mostrano simmetrie utili per i calcoli cinematici (polso, gomito, etc.). Le stesse simmetrie possono essere usate per creare uno spazio discreto in 4D, non necessariamente cilindrico, che funzioni con questo algoritmo.

**Altri algoritmi di ricerca di percorso.** L'intelligenza artificiale, in qualità di disciplina relativamente matura, ha prodotto una varietà di algoritmi di ricerca di percorso che possono essere applicati alla ricerca sullo spazio discreto in 4D proposto in questo articolo scientifico. In aggiunta a questi, la robotica ha prodotto altri algoritmi specifici che sono più appropriati per i movimenti dei robot industriali. Qualsiasi algoritmo di ricerca di percorso che sia capace di generare un percorso tracciabile in uno spazio a quattro dimensioni può essere un buon candidato per ulteriore ricerca.

**Analisi dei controllori dei robot industriali.** Mentre Matlab® garantisce ad un utente di essere accurato nel produrre i punti di via per il tracciamento della traiettoria con i valori dei giunti tramite il controllore ABB, non è parte della simulazione effettuata per questo articolo scientifico il tracciamento della *performance* del controllore ABB stesso o di altri controllori robotici. La traiettoria risultante eseguita in RobotStudio® appare visibilmente veloce e regolare, ma non ci sono funzionalità incluse in RobotStudio® per analizzare ed esportare la traiettoria tracciata quando il controllore è pilotato da istruzioni in linguaggio RAPID. Sarebbe interessante poter analizzare nel futuro lavoro il metodo di controllo di alcuni

controllori robotici industriali e comparare la stabilità, accuratezza e rapidità di tracciamento di una traiettoria.

## RINGRAZIAMENTI

Questo articolo scientifico è stato tradotto in onore e per gratitudine a Carlo de Giorgio e Roberta Eller Vainicher.

## RIFERIMENTI BIBLIOGRAFICI

- [1] L. Zhao, J. Zhao, H. Liu, D. Manocha, Collision-Free Kinematics for Redundant Manipulators in Dynamic Scenes using Optimal Reciprocal Velocity Obstacles, (2018). <http://arxiv.org/abs/1811.00600> (accessed July 1, 2020).
- [2] A.G. Jiokou Kouabon, A. Melingui, J.J.B. Mvogo Ahanda, O. Lakhal, V. Coelen, M. KOM, R. Merzouki, A Learning Framework to inverse kinematics of high DOF redundant manipulators, Mech. Mach. Theory. 153 (2020) 103978. <https://doi.org/10.1016/j.mechmachtheory.2020.103978>.
- [3] S. Alartsev, S. Stellmacher, F. Ortmeier, Robotic Task Sequencing Problem: A Survey, J. Intell. Robot. Syst. Theory Appl. 80 (2015) 279–298. <https://doi.org/10.1007/s10846-015-0190-6>.
- [4] Z. Aljarboua, Geometric path planning for general Robot Manipulators, in: Proc. World Congr. Eng. Comput. Sci., 2009: pp. 20–22.
- [5] A. Nemati, M. Kumar, Modeling and control of a single axis tilting quadcopter, in: Proc. Am. Control Conf., Institute of Electrical and Electronics Engineers Inc., 2014: pp. 3077–3082. <https://doi.org/10.1109/ACC.2014.6859328>.
- [6] S. Kūcūk, Z. Bingül, The inverse kinematics solutions of industrial robot manipulators, in: Proc. IEEE Int. Conf. Mechatronics 2004, ICM'04, 2004: pp. 274–279. <https://doi.org/10.1109/icmech.2004.1364451>.
- [7] M.H. Raibert, B.K.P. Horn, Manipulator control using the configuration space method, Ind. Rob. 5 (1978) 69–73. <https://doi.org/10.1108/eb004494>.
- [8] P. Tavares, J. Lima, P. Costa, A.P. Moreira, Multiple manipulators path planning using double A\*, Ind. Rob. 43 (2016) 657–664. <https://doi.org/10.1108/IR-01-2016-0006>.
- [9] T. Rybus, Obstacle avoidance in space robotics: Review of major challenges and proposed solutions, Prog. Aerosp. Sci. 101 (2018) 31–48. <https://doi.org/10.1016/j.paerosci.2018.07.001>.
- [10] C. Kohrt, R. Stamp, A.G. Pipe, J. Kiely, G. Schiedermeier, An online robot trajectory planning and programming support system for industrial use, Robot. Comput. Integr. Manuf. 29 (2013) 71–79. <https://doi.org/10.1016/j.rcim.2012.07.010>.
- [11] Y. Yu, V. Lippiello, 6D pose task trajectory tracking for a class of 3D aerial manipulator from differential flatness, IEEE Access. 7 (2019) 52257–52265. <https://doi.org/10.1109/ACCESS.2019.2910379>.
- [12] A.M. Kabir, A. Kanyuck, R.K. Malhan, A. V. Shembekar, S. Thakar, B.C. Shah, S.K. Gupta, Generation of synchronized configuration space trajectories of multi-robot systems, in: Proc. - IEEE Int. Conf. Robot. Autom., Institute of Electrical and Electronics Engineers Inc., 2019: pp. 8683–8690. <https://doi.org/10.1109/ICRA.2019.8794275>.
- [13] M. Bergerman, Y. Xu, Planning collision-free motions for underactuated manipulators in constrained configuration space, in: Proc. - IEEE Int. Conf. Robot. Autom., IEEE, 1997: pp. 549–555. <https://doi.org/10.1109/robot.1997.620094>.
- [14] Y. Han, W. Zhao, J. Pan, Z. Ye, R. Yi, Y.-J. Liu, A Configuration-Space Decomposition Scheme for Learning-based Collision Checking, (2019). <http://arxiv.org/abs/1911.08581> (accessed July 1, 2020).
- [15] S. Kumar, S. Choudhary, S. Srinivasa, Learning Configuration Space Belief Model from Collision Checks for Motion Planning,

- (2019). <http://arxiv.org/abs/1901.07646> (accessed July 1, 2020).
- [16] O. Salzmann, K. Solovey, D. Halperin, Motion Planning for Multilink Robots by Implicit Configuration-Space Tiling, *IEEE Robot. Autom. Lett.* 1 (2016) 760–767. <https://doi.org/10.1109/LRA.2016.2524066>.
- [17] K. Wei, B. Ren, A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm, *Sensors*. 18 (2018) 571. <https://doi.org/10.3390/s18020571>.
- [18] R. Geraerts, M.H. Overmars, A comparative study of probabilistic roadmap planners, in: *Springer Tracts Adv. Robot.*, Springer, Berlin, Heidelberg, 2004: pp. 43–57. [https://doi.org/10.1007/978-3-540-45058-0\\_4](https://doi.org/10.1007/978-3-540-45058-0_4).
- [19] L.E. Kavvaki, P. Švestka, J.C. Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Trans. Robot. Autom.* 12 (1996) 566–580. <https://doi.org/10.1109/70.508439>.
- [20] V.D. Tourassis, M.H. Ang, Identification and Analysis of Robot Manipulator Singularities, *Int. J. Rob. Res.* 11 (1992) 248–259. <https://doi.org/10.1177/027836499201100307>.
- [21] E. Fernandes, P. Costa, J. Lima, G. Veiga, Towards an orientation enhanced astar algorithm for robotic navigation, in: *Proc. IEEE Int. Conf. Ind. Technol.*, Institute of Electrical and Electronics Engineers Inc., 2015: pp. 3320–3325. <https://doi.org/10.1109/ICIT.2015.7125590>.
- [22] J. Yao, C. Lin, X. Xie, A.J.A. Wang, C.C. Hung, Path planning for virtual human motion using improved A\* algorithm, in: *ITNG2010 - 7th Int. Conf. Inf. Technol. New Gener.*, 2010: pp. 1154–1158. <https://doi.org/10.1109/ITNG.2010.53>.
- [23] H. Wang, J. Zhou, G. Zheng, Y. Liang, HAS: Hierarchical a-star algorithm for big map navigation in special areas, in: *Proc. - 2014 Int. Conf. Digit. Home, ICDH 2014*, Institute of Electrical and Electronics Engineers Inc., 2014: pp. 222–225. <https://doi.org/10.1109/ICDH.2014.49>.
- [24] M. Noto, H. Sato, Method for the shortest path search by extended Dijkstra algorithm, in: *Proc. IEEE Int. Conf. Syst. Man Cybern.*, IEEE, 2000: pp. 2316–2320. <https://doi.org/10.1109/icsmc.2000.886462>.
- [25] A.S. Vázquez, A. Adán, Nonprobabilistic anytime algorithm for high-quality trajectories in high-dimensional spaces, *Robotica*. 30 (2012) 289–303. <https://doi.org/10.1017/S0263574711000506>.
- [26] M. Kelemen, I. Virgala, T. Lipták, E. Miková, F. Filakovský, V. Bulej, A Novel Approach for a Inverse Kinematics Solution of a Redundant Manipulator, *Appl. Sci.* 8 (2018) 2229. <https://doi.org/10.3390/app8112229>.
- [27] X. Xu, Y. Hu, J. Zhai, L. Li, P. Guo, A novel non-collision trajectory planning algorithm based on velocity potential field for robotic manipulator, *Int. J. Adv. Robot. Syst.* 15 (2018) 172988141878707. <https://doi.org/10.1177/1729881418787075>.
- [28] M. Kallmann, M. Mataric, Motion planning using dynamic roadmaps, in: *Proc. - IEEE Int. Conf. Robot. Autom.*, 2004: pp. 4399–4404. <https://doi.org/10.1109/robot.2004.1302410>.
- [29] Y. Ansari, E. Falotico, Y. Mollard, B. Busch, M. Cianchetti, C. Laschi, A Multiagent Reinforcement Learning approach for inverse kinematics of high dimensional manipulators with precision positioning, in: *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics*, IEEE Computer Society, 2016: pp. 457–463. <https://doi.org/10.1109/BIOROB.2016.7523669>.
- [30] M. Ossenkopf, P. Ennen, R. Vossen, S. Jeschke, Reinforcement Learning for Manipulators without Direct Obstacle Perception in Physically Constrained Environments, *Procedia Manuf.* 11 (2017) 329–337. <https://doi.org/10.1016/j.promfg.2017.07.115>.
- [31] T. Li, S. Sun, Y. Gao, Fan-shaped grid based global path planning for mobile robot, *Jiqiren/Robot.* 32 (2010) 547–552. <https://doi.org/10.3724/SP.J.1218.2010.00547>.
- [32] C. Llopis-Albert, F. Rubio, F. Valero, Optimization approaches for robot trajectory planning, *Multidiscip. J. Educ. Soc. Technol. Sci.* 5 (2018) 1. <https://doi.org/10.4995/muse.2018.9867>.
- [33] K. Kaltsoukalas, S. Makris, G. Chryssolouris, On generating the motion of industrial robot manipulators, *Robot. Comput. Integr. Manuf.* 32 (2015) 65–71. <https://doi.org/10.1016/j.rcim.2014.10.002>.
- [34] L. Sciavicco, B. Siciliano, *Modelling and Control of Robot Manipulators*, Springer London, London, 2000. <https://doi.org/10.1007/978-1-4471-0449-0>.
- [35] D.L. Peiper, *The Kinematics of Manipulators Under Computer Control*, Stanford University CA, 1968. <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=AD0680036>.
- [36] P. Reynoso-Mora, W. Chen, M. Tomizuka, On the time-optimal trajectory planning and control of robotic manipulators along predefined paths, in: *Proc. Am. Control Conf.*, 2013: pp. 371–377. <https://doi.org/10.1109/acc.2013.6579865>.
- [37] Z. Shiller, Off-line and on-line trajectory planning, in: *Mech. Mach. Sci.*, Kluwer Academic Publishers, 2015: pp. 29–62. [https://doi.org/10.1007/978-3-319-14705-5\\_2](https://doi.org/10.1007/978-3-319-14705-5_2).
- [38] T. Flash, R.B. Potts, Communication, *Int. J. Rob. Res.* 7 (1988) 48–57. <https://doi.org/10.1177/027836498800700505>.
- [39] H.H. Tan, R.B. Potts, A discrete path/trajectory planner for robotic arms, *J. Aust. Math. Soc. Ser. B. Appl. Math.* 31 (1989) 1–28. <https://doi.org/10.1017/s0334270000006457>.
- [40] A. Stentz, The focussed D\* algorithm for real-time replanning, in: *IJCAI'95 Proc. 14th Int. Jt. Conf. Artif. Intell. - Vol. 2*, 1995: pp. 1652–1659. <https://dl.acm.org/doi/abs/10.5555/1643031.1643113> (accessed June 21, 2020).
- [41] B.H. Lee, C.S.G. Lee, Collision-Free Motion Planning of Two Robots, *IEEE Trans. Syst. Man Cybern.* 17 (1987) 21–32. <https://doi.org/10.1109/TSMC.1987.289330>.



Andrea de Giorgio è dottorando in ingegneria della produzione al KTH Istituto Reale di Tecnologia a Stoccolma, Svezia. Andrea è nato a Napoli, Italia, nel 1987. Ha conseguito una laurea in ingegneria elettronica dall'Università degli Studi di Napoli Federico II, Italia, nel 2013 e una laurea magistrale in ingegneria informatica con specializzazione in machine learning dal KTH Istituto Reale di Tecnologia a Stoccolma, Svezia, nel 2015. I suoi interessi di ricerca includono intelligenza artificiale, machine learning e algoritmi di deep learning per processi di conoscenza industriali di collaborazione uomo-macchina, in ambienti di realtà, virtuale aumentata o mista.



Lihui Wang è Professore e Direttore di manifattura sostenibile al KTH Istituto Reale di Tecnologia a Stoccolma, Svezia. Lihui è attivo in varie attività professionali. Ha pubblicato nove libri e in eccesso 500 pubblicazioni scientifiche. I suoi interessi di ricerca sono incentrati sui sistemi ciberfisici, la manifattura su cloud, controllo e monitoraggio in tempo reale, manutenzione predittiva, collaborazione uomo-robot e sistemi di manifattura sostenibile adattivi. È membro della associazione degli ingegneri canadese, del CIRP, SME ed ASME. È anche Ingegnere Professionista in Canada, Presidente della North American Manufacturing Research Institution of SME, e Presidente della Swedish Production Academy. È Editore Capo del International Journal of Manufacturing Research, di Robotics and Computer-Integrated Manufacturing, e del Journal of Manufacturing Systems.