UNIVERSITÀ DEGLI STUDI DI PADOVA

—

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

—

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA DELL'AUTOMAZIONE

MODELLAZIONE TRIDIMENSIONALE DI CELLE ROBOTIZZATE FLESSIBILI

RELATORE: CH.MO PROF. ING. ALDO ROSSI
CORRELATORE: CH.MO PROF. ING. GIULIO ROSATI

LAUREANDO: MURARO RICCARDO

ANNO ACCADEMICO 2013-2014

Indice

Sommario

La tesi riguarda lo studio e realizzazione di un sistema di collision detection per simulazione robotica compatibile con qualsiasi modello CAD di robot a 6 o 4 gradi di libertà. L'implementazione tridimensionale in MATLAB riguarda un'interfaccia grafica che consente la realizzazione del modello del robot mediante nuvola di punti e/o modello geometrico al fine di poter essere utilizzato nel test di collisione fra robot e ambiente.

Introduzione

La tesi che viene descritta nelle pagine che seguono si inserisce nel contesto della sicurezza attiva dei robot, che ha come obiettivo quello di evitare collisioni accidentali fra un manipolatore robotizzato e ambiente circostante costituendo un controllo di movimento preventivo gestito via software.

Per lo svolgimento del progetto è stato utilizzato il software di programmazione MATLAB, acronimo di Matrix Laboratory. La sua specializzazione nel calcolo numerico e matriciale è stata ampiamente sfruttata per la realizzazione del sistema di collision detection più appropriato. D'altro canto il limite maggiore di MATLAB è dovuto alla complessità del sistema di gestione grafica e rappresentazione tridimensionale. In questo progetto si è cercato di limare questa deficienza ricercando degli espedienti che potessero rendere questo software adatto anche alla simulazione e modellazione tridimensionale di sistemi robotizzati.

Capitolo 1

Collision Detection

L'argomento riguardante collision detection è molto studiato e apparentemente sembra un facile problema, ovvero determinare se due (o più) oggetti sono in collisione. Più precisamente collision detection consiste nel problema di determinare *se*, *quando* e *dove* due oggetti entrano in contatto. "Se" implica stabilire il risultato Booleano, vero o falso, dell'esito dell'intersezione fra gli oggetti. "Quando" determina il tempo affinché avvenga la collisione durante il movimento. "Dove" stabilisce come gli oggetti stanno entrando in contatto. Si può affermare che queste tre tipologie di interrogazione crescono computazionalmente in complessità secondo l'ordine dato. Nel determinare quando e dove si fa più propriamente riferimento al termine *collision determination*, mentre *intersection detection* e *interference detection* sono sinonimi che possono essere usati con collision detection.

Le risposte alle tre tipologie di interrogazione (se, dove, quando) vanno ricercate nella stessa branca dell'informatica denominata *geometria computazionale* che studia le strutture dati e gli algoritmi efficienti per la soluzione di problemi di natura geometrica e la loro implementazione al calcolatore.

Collision detection è di fondamentale importanza in molte applicazioni, fra cui computer graphics, virtual prototyping, robotica, simulazioni ingegneristiche (per nominarne alcuni). Nell'ambito della robotica il collision detection è il principale aspetto che riguarda il concetto più astratto e generale di sicurezza attiva dei robot, ovvero evitare collisioni fra robot e ostacoli presenti all'interno dello spazio di lavoro.

Utilizzando una metafora, progettare un efficiente sistema di collision detection è come come realizzare un puzzle: i tasselli devono essere collegati insieme a formare dei particolari del puzzle senza perdere il contesto generale dell'immagine affinché cominci ad apparire nella sua totalità.

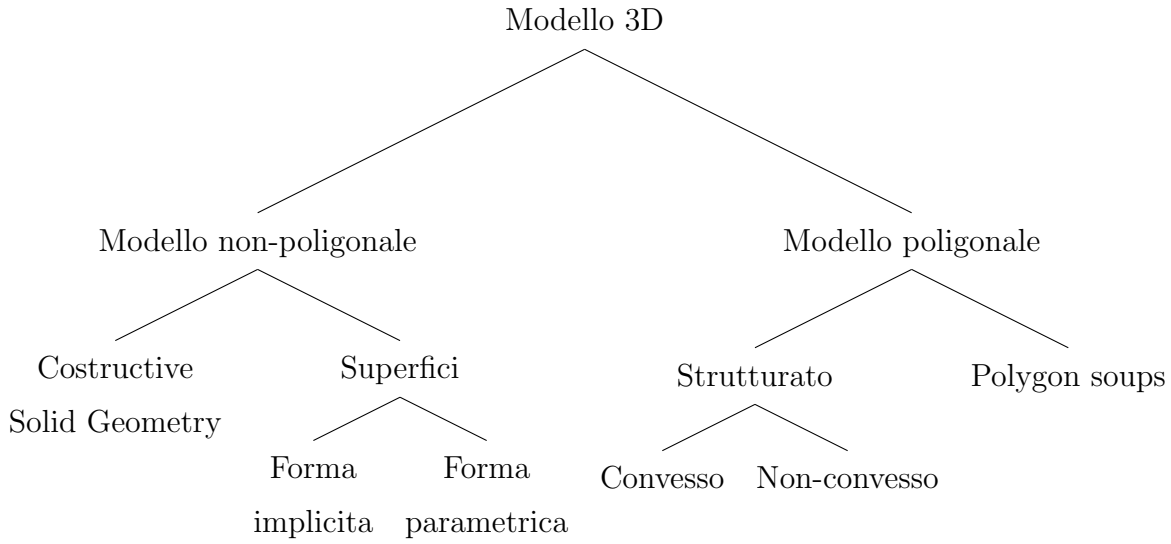
Ci sono diversi fattori da prendere in considerazione per realizzare un sistema di collision detection:

1. Rappresentazione del modello
2. Tipologia di interrogazione
3. Caratteristiche dell'ambiente
4. Prestazioni e robustezza
5. Facilità di implementazione e utilizzo

Nelle prossime sezioni vengono analizzati con maggiore dettaglio questi aspetti e in quale modo sono in relazione fra di loro.

1.0.1 Rappresentazione del modello

La struttura di un algoritmo di collision detection dipende anzitutto dalla rappresentazione del modello. Una possibile tassonomia riguarda la suddivisione fra modelli poligonali e non-poligonali, adottata in [1].



La rappresentazione di oggetti più utilizzata riguarda modelli descritti da una superficie formata da una collezione di poligoni. Esistono molti algoritmi di collision detection che fanno riferimento a questa struttura. La classe più generale dei modelli poligonali è *polygon soup*, ovvero una collezione di poligoni senza nessuna informazione sulla loro topologia e come essi siano connessi fra loro. Se i poligoni sono strutturati a formare un solido (poliedro) tramite superficie chiusa, è possibile definire la regione interna ed esterna del modello. In questo casi si assume che il modello costituisca una varietà topologica[1] chiusa ed altre proprietà addizionali riguardanti la convessità della struttura possono essere sfruttate efficacemente negli algoritmi di collision detection.

La rappresentazione mediante CSG, acronimo di *Costructive Solid Geometry*, sono modelli formati a partire da primitive geometriche elementari quali blocchi, sfere, cilindri, coni e tori, combinati fra loro a formare geometrie complesse attraverso operazioni di unione, intersezione e differenza.

Le superfici possono essere rappresentate da una funzione in forma implicita $f : \Re^3 \mapsto \Re$ definendo univocamente l'interno $f(x, y, z) < 0$ ed esterno $f(x, y, z) > 0$ del modello. Le funzioni polinomiali e forme quadratiche sono utilizzate spesso per rappresentare le primitive degli oggetti CSG in quanto in generale costituiscono una varietà topologica chiusa.

Le superfici in forma parametrica sono funzioni che mappano dal piano allo spazio $f : \Re^2 \mapsto \Re^3$. Diversamente dalle superfici in forma implicita e rappresentazioni CSG, le superfici parametriche non costituiscono una varietà topologica chiusa, ovvero esse non rappresentano dei modelli solidi, ma d'altro canto sono in grado di descrivere efficacemente una superficie piana.

1.0.2 Tipologia di interrogazione

Nel caso più semplice si vuole conoscere se due modelli sono in collisione, ovvero l'esito dell'algoritmo di collision detection assume il carattere di predicato

[1] Una varità topologica, manifold in inglese, è uno spazio topologico in cui ogni punto ha un intorno aperto che è omeomorfo ad uno spazio euclideo $\Re^n$. Questo concetto che si incontra spesso nell'ambito della geometria computazionale, riguarda sostanzialmente delle buone proprietà geometriche dello spazio topologico.

Booleano. In alcuni casi può essere necessario conoscere quali parti (se esistono) che collidono, per esempio trovare la loro intersezione. Altre volte invece si vuole conoscere la loro separazione: distanza Euclidea minima fra gli oggetti. Quando gli oggetti sono in movimento può anche essere importante conoscere l'istante temporale per il quale avviene la collisione.

In questo lavoro si vuole rispondere essenzialmente al primo quesito, ovvero determinare se avviene la collisione ed eventualmente il tempo impiegato per determinarla al fine di poter valutare l'efficienza dell'algoritmo.

1.0.3 Caratteristiche dell'ambiente

Esistono dei parametri del contesto applicativo da considerare nella progettazione e scelta dell'algoritmo più appropriato di collision detection. Questi sono: numero di oggetti, movimento degli oggetti, tipologia degli oggetti.

Numero di oggetti

Se il problema consiste solo in una coppia di oggetti, si parla di *pair-processing*. Se ci sono differenti parti si parla di *n-body processing*. Intuitivamente una simulazione con n oggetti che richiede un test per ogni coppia di oggetti, si giunge a complessità $O(n^2)$ nel caso peggiore. Ciò che spesso si usa fare a questo proposito è ridurre il numero di coppie da verificare separando il test di collisione di molti oggetti in due fasi dette *broad-phase* e *narrow-phase*.

La broad phase identifica gruppi di oggetti che *potrebbero* collidere e velocemente determina se essi sicuramente non stanno collidendo. La narrow phase costituisce il pair-processing fra questi sottogruppi. In questo modo è possibile diminuire significativamente il carico computazionale rispetto all'approccio naif di confrontare tutte le coppie di oggetti.

Movimento degli oggetti

Il comportamento reale degli oggetti è quello di muoversi *simultaneamente* durante un assegnato periodo temporale con ogni eventuale collisione risolta all'interno

di questo step di tempo. Esistono diversi approcci che permettono di risolvere il problema in un ambiente dinamico.

La scelta che spesso viene adottata è quella di suddividere l'intervallo in passi temporali e verificare la collisione ad ogni step come se gli oggetti fossero stazionari alla propria posizione e velocità nulla. Questa *discretizzazione* solitamente comporta un minor carico computazionale rispetto a considerare l'intero movimento continuo, al costo di considerare un passo temporale sufficientemente piccolo.
Spesso gli oggetti sono soggetti a trasformazioni rigide di rotazione e traslazione. In questi casi la relazione geometrica fra un'istante temporale e il precedente può differire di poco e il movimento relativo essere piccolo. Un algoritmo con *coerenza temporale* potrebbe utilizzare a proprio vantaggio questo fatto per ridurre ulteriormente la broad phase.

Un alternativa potrebbe essere quella di considerare il volume occupato nel movimento continuo nell'intervallo di tempo, chiamato *swept volume*. Se lo swept volume di due oggetti non interseca, non c'è sicuramente collisione. Tuttavia se i volumi intersecano, non è condizione sufficiente in quanto potrebbero non aver colliso durante il movimento.

In ogni caso è sempre preferibile considerare un oggetto statico rispetto ad un altro in quanto semplifica l'analisi del moto che potrebbe essere un'operazione onerosa. Se questa situazione non si verifica in pratica, è sufficiente considerare il moto relativo fra gli oggetti, considerandone uno come fosse un oggetto statico.

Tipologia degli oggetti

Ci sono due aspetti che riguardano la tipologia degli oggetti.

Il primo aspetto riguarda la loro *dimensione* reciproca. Questo può influire significativamente sulla quantità di verifiche di collisione che vengono eseguite. La broad phase in presenza di oggetti di dimensione non-omogenea dovrebbe essere fatta in modo più sofisticato.

Il secondo aspetto potrebbe riguardare la presenza di *deformazione* degli oggetti durante il movimento. Questo comporterebbe un sistema di collision detection molto più sofisticato rispetto al caso in cui gli oggetti siano rigidi.

1.0.4 Prestazioni e robustezza

Chiaramente dal punto di vista delle prestazioni è molto importante ottimizzare il tempo medio di risposta al collision detection. Allo stesso modo è importante assicurarsi che il caso peggiore non comporti un ordine di grandezza maggiore rispetto al caso medio.

Le performance sono un aspetto da considerare anche nella fase di progettazione del sistema di collision detection oltre che nella fase di ottimizzazione. Di sicuro rendere efficiente l'elaborazione della broad phase, oppure utilizzare algoritmi con coerenza temporale, possono portare ad un risparmio computazionale e quindi minor tempo di risposta.

Dal punto di vista delle prestazioni è importante distinguere le operazioni da svolgere real-time e quelle off-line. Nel primo caso l'efficienza temporale è essenziale al fine di rendere trasparente il carico computazionale dovuto al collision detection. Capita spesso che queste specifiche temporali non possano essere soddisfatte real-time o che richiedano strumenti con potenza di calcolo troppo elevate. In questi casi, e in generale, la soluzione migliore è quella di rendere ove possibile off-line il processo sfruttando al meglio le risorse disponibili.

La robustezza dell'algoritmo, ovvero la capacità del di operare computazioni numeriche e configurazioni geometriche che sono complicate da gestire fornendo un risultato corretto e/o aspettato, è un aspetto importante in uno studio sperimentale (e non solo). In generale, il problema della robustezza può essere suddiviso in due classi: *robustezza numerica* e *robustezza geometrica*. Il primo concetto riguarda la precisione sull'uso delle variabili e la gestione di errori numerici quali per esempio errori di approssimazione. Il secondo assicura la correttezza topologica e la consistenza geometrica, per esempio un triangolo che degenera in un punto o un poligono i cui vertici non giacciono sullo stesso piano.

1.0.5 Implementazione e facilità di utilizzo

Nell'implementazione di un sistema di collision detection possono essere molto importanti anche l'insieme delle caratteristiche desiderate, in relazione con il tempo di sviluppo necessario alla realizzazione di esse che in genere costituiscono

dei casi particolari. Possono insorgere diversi problemi dovuti a: il carattere di generalità dell'algoritmo, la capacità di operare con oggetti di diversa misura, il tempo impiegato per la ricostruzione legata alla struttura dati, ecc.

In pratica un buon progetto si misura in base alle caratteristiche desiderate, e la facilità di utilizzo del sistema va misurata da quanto il sistema è in grado di scostarsi dal comportamento standard.

Le due caratteristiche che si richiedono in generale sono: la capacità di gestire problemi in modo autonomo o automatico e rendere il sistema il più possibile flessibile e portabile.

1.1 Bounding volumes

Verificare direttamente la collisione fra la geometrie di due oggetti può essere oneroso, soprattutto quando gli oggetti consistono in centinaia o migliaia di poligoni. Per minimizzare questo costo, si costruisce un volume semplice, detto *bounding volume* (BV), che incapsula uno o più oggetti di natura più complessa. In questo modo è possibile avere un test di intersezione molto più veloce rispetto alla geometria complessa dell'oggetto che racchiudono.

Bounding volume vengono utilizzati principalmente per avere una rapida prova di non-collisione, evitando test successivi fra geometrie complesse. Questo comporta generalmente un significativo miglioramento delle prestazioni. Per alcune applicazioni, la verifica di intersezione fra BV è una prova sufficiente per la collisione.

Il problema inizialmente può essere considerato *omogeneo*, ovvero test di collisione fra BV rappresentati della stessa forma. Tuttavia non è poco comune utilizzare tipologie *eterogenee* di BV cosicché è necessario generalizzare test di intersezione fra forme diverse.

1.1.1 Caratteristiche di un BV

Non tutte le forme geometriche possono effettivamente costituire un BV. Un buon candidato per bunding volume dovrebbe possedere le seguenti caratteristiche:

- test di collisione non oneroso
- volume aderente all'oggetto (tight-fitting)
- facilità di aggiornamento (rotazione e traslazione)
- utilizzare poca memoria

Per supportare un test di intersezione non oneroso, un BV deve essere una forma geometrica semplice. D'altro parte il volume deve ricoprire nel modo più aderente possibile l'oggetto, costituendo un trade-off fra fitting e costo del test. L'ultima caratteristica in quest'ambito non ricopre un ruolo importante.

Un test di collisione comporta anche ad interrogazioni aggiuntive con primitive geometriche come, ad esempio, inclusione di un punto, intersezione di rette con volumi, intersezioni di piani con poliedri ecc.

I bounding volume sono tipicamente elaborati e creati in una fase di prepocessing (offline) invece che runtime (inline). Molto spesso è necessario riallineare i BV quando essi contengono degli oggetti mobili. Questa operazione potrebbe essere onerosa e talvolta è conveniente ri-computare ricreando il BV dall'inizio.

In generale, queste caratteristiche desiderabili per un bounding volume sono mutualmente esclusive, e non esiste un specifico BV migliore per ogni applicazione.

1.1.2 Tipologie di BV

La figura 1.1 illustra un esempio di cinque comuni tipologie di bounding volume utilizzate. L'ordine rispecchia il miglior fitting e selettività in contrapposizione con la velocità del test e miglior utilizzo della memoria.

Il libro di Ericson [2] fornisce una casistica completa di algoritmi utili per realizzare il test di collisione fra i principali bounding volume esistenti e come risolvere in modo efficiente molti problemi di geometria computazionale che possono verificarsi in fase di realizzazione.

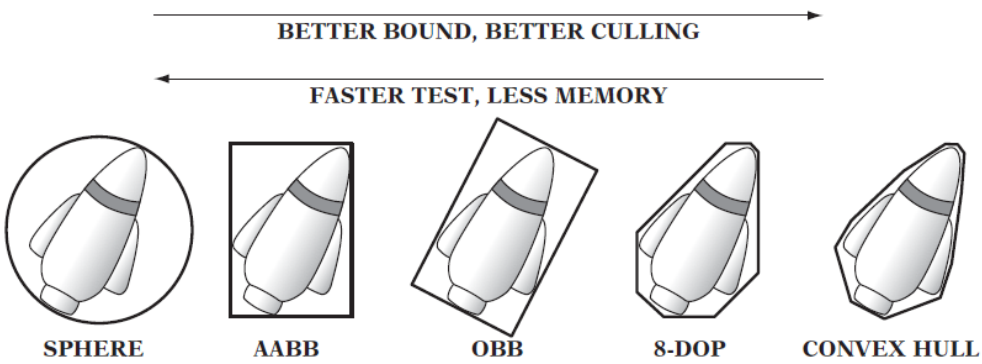
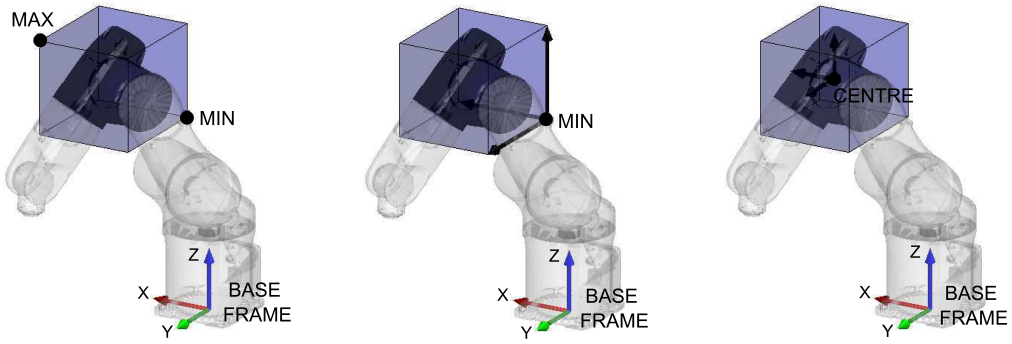


Figura 1.1: Tipologie di BV: sfere, axis-aligned bounding box (AABB), oriented bounding box (OBB), eight-direction discrete orientation polytope (8-DOP), e convex hull (inviluppo convesso)

Le forme di bounding volume nello spazio 3D più utilizzate sono:

- Axis-aligned bounding boxes (AABBs)
- Oriented bounding boxes (OBBs)
- Sphere-swept volumes (SSVs)
  - Sphere-swept Point (SSP)
  - Shpere-swept Line (SSL)
  - Sphere-swept Rectangle (SSR)
- Halfspace Intersection Volumes
  - Slab-based Volumes
  - Discrete-orientation Polytopes (k-DOPs)
  - Approximate Convex Hull

Gli AABBs sono un parallelepipedo caratterizzato da sei facce che sono sempre parallele con gli assi di dato sistema di coordinate di riferimento. Il principale vantaggio è che il test di collisione involve nel confronto diretto di valori coordinate individuali.



(a) Due punti: minimo e massimo

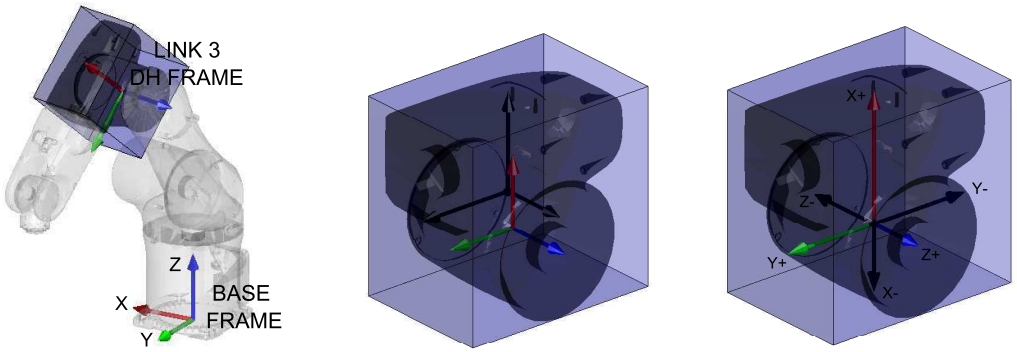
(b) Un punto e 3 lunghezze lati

(c) Punto centrale e 3 semi-lunghezze dei lati

Figura 1.2: Possibili rappresentazioni di AABB.

Le modalità con le quali è possibile rappresentare un AABB sono rappresentate in figura 1.2. L'esempio mostra come sia possibile rappresentare il link di un robot, considerando come sistema di riferimento la terna della base robot.

Risulta evidente come gli AABB comportino un volume eccessivamente grande rispetto all'oggetto che approssimano. Inoltre si rende necessario aggiornarne le dimensioni per ogni configurazione del robot. Tale operazione può risultare onerosa.



(a) Terna di DH del link

(b) Punto centrale e 3 semi-lunghezze dei lati

(c) 6 lunghezze dei lati

Figura 1.3: Esempi di rappresentazioni OBB riferite alla terna di DH del link.

Gli OBBs differiscono dagli AABBs in quanto possono avere un arbitraria orientazione degli assi. Questo consente un miglior fitting dell'oggetto anche se il test di collisione risulta più complesso. La scelta più ovvia per quanto riguarda la

rappresentazione di un link robot, è quella di utilizzare come riferimento degli assi la terna di Denavith Hartemberg (figura 1.3a). Il questo modo è sufficiente definire una sola volta l'OBB e non è necessario aggiornarne le dimensioni a seguito di un cambio di configurazione del robot. Le possibili rappresentazioni (figura 1.3b e 1.3c) differiscono da un AABB per l'aggiunta di trasformazione di coordinate data dalla matrice di rototraslazione di DH.

La sfera assicura il più veloce collision test in quanto è invariante a rotazioni; è sufficiente una traslazione del centro nella nuova posizione. Per contro non costituisce una buona rappresentazione dell'oggetto. La possibile soluzione è di utilizzare le sfere in combinazione con segmenti, rettangoli o parallelepipedi (figura 1.4).

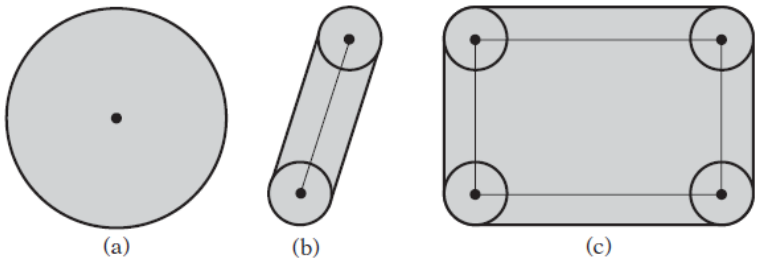
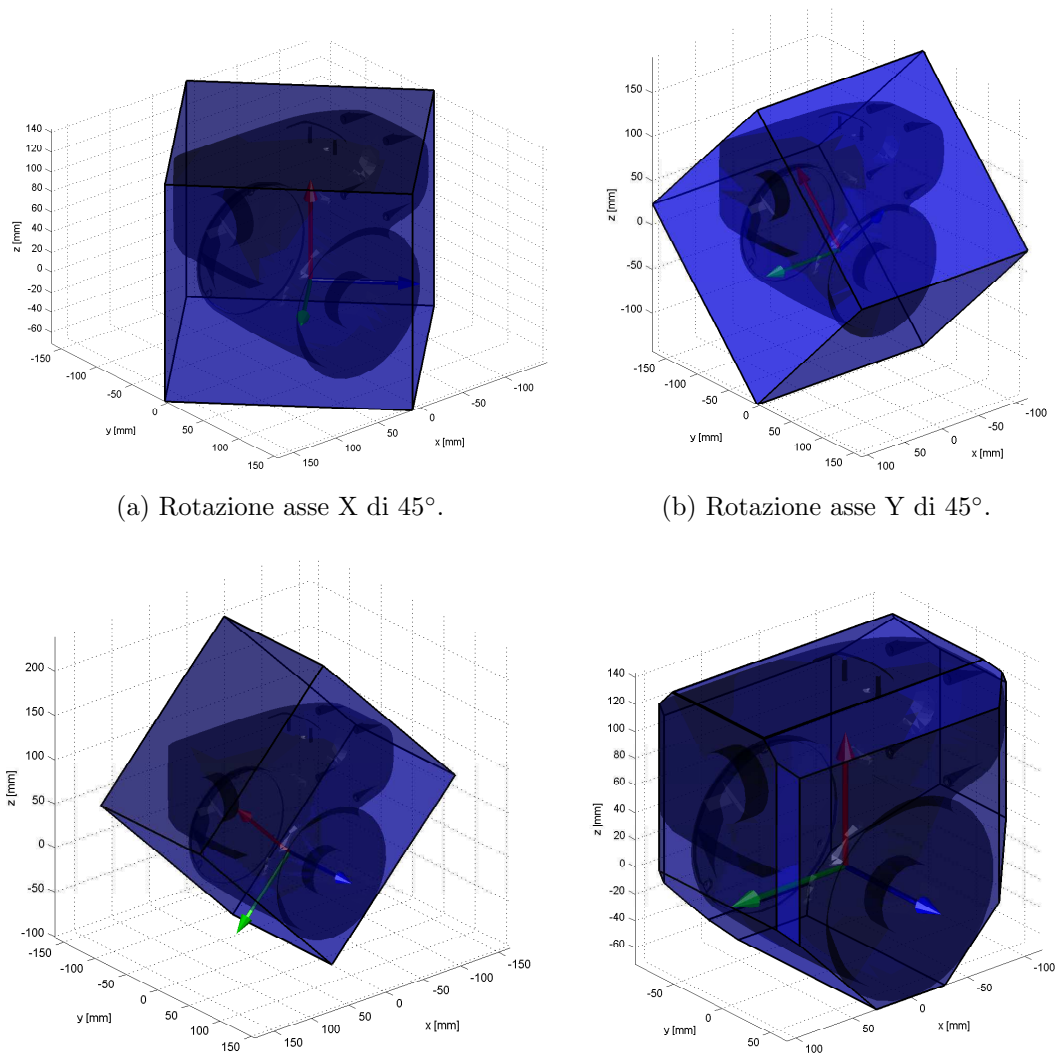


Figura 1.4: Tipologie di SSV: (a) Sphere-swept Point (SSP), (b) Shpere-swept Line (SSL), (c) Sphere-swept Rectangle (SSR)

Spesso si verifica che i BV siano rappresentati da poliedri convessi. Sono quindi definiti dall'intersezione di un'insieme di piani, ognuno dei quali determina un semispazio aperto, e l'intersezione di essi forma un volume chiuso nello spazio tridimensionale. Gli AABBs e OBBs sono un caso particolare: corrispondono all'intersezione di sei semispazi.

Generalmente più semispazi sono utilizzati e migliore risulterà la rappresentazione dell'oggetto in termini di fit. Un possibile approccio è stato fornito da Kay e Kajiya [3] i quali hanno rappresentato il bounding volume come intersezione di slab (regione infinita compresa fra due piani). Basato su questo principio esiste una categoria di BV denominati discrete-orientation polytopes (k-DOPs) definiti da un'insieme finito e fissato di k slab le cui normali sono relative a k diverse e fissate orientazioni del sistema di riferimento.



(a) Rotazione asse X di 45°.

(b) Rotazione asse Y di 45°.

(c) Rotazione asse Z di 45°.

(d) Intersezione dei Box.

Figura 1.5: Esempio di 18-DOP per un link robot.

Un'importante categoria sono i 18-DOP che sono definiti dall'intersezione di piani aventi 18 normali diverse nello spazio tridimensionale. Corrisponde ad un OBB avente i 12 lati tagliati. Per ottenerlo è sufficiente ruotare uno per volta i 3 assi del sistema di riferimento come mostrato in figura 1.5.

k-DOPs comportano (probabilmente) ad un miglior fitting degli OBBs, certamente al crescere di k il k-DOP approssima sempre meglio la convoluzione convessa che è l'ottimo in termini di fit per poliedri convessi.

1.1.3 Bounding Volume Hierarchies

Una possibile strategia per BV consiste nel rappresentare tali forme in una gerarchia ad albero chiamata *bounding volume hierarchy* (BVH). La complessità computazionale può essere ridotta di un ulteriore fattore logaritmico.

Il principio è di attribuire ai nodi foglia l'insieme originale dei BV (che possono contenere una o più primitiva di base), e raggruppare questi BV in modo ricorsivo in altri BV più grandi che li contengono. Il risultato è una struttura ad albero dove la radice racchiude la struttura degli oggetti in un singolo bounding volume.

Questo comporta che il test di collisione dei nodi figli non avviene se i volumi dei suoi parenti non intersecano. É importante osservare che BV dei nodi parenti non racchiudono necessariamente i BV dei nodi figli. In generale il bounding volume dei parenti si costruisce sulle primitive contenute nel sottoalbero dei figli.

Sebbene sia sempre possibile creare manualmente questa gerarchia ad albero, la soluzione migliore è quella di automatizzare il processo. Un progettista potrebbe teoricamente aggiustare manualmente in modo migliore i volumi per adattarli alla collisione, ma questo comporta un effettivo dispendio di tempo. La soluzione automatizzata incorre necessariamente a dei sotto-problemi, ma potrebbe condurre a dei risultati comunque soddisfacenti.

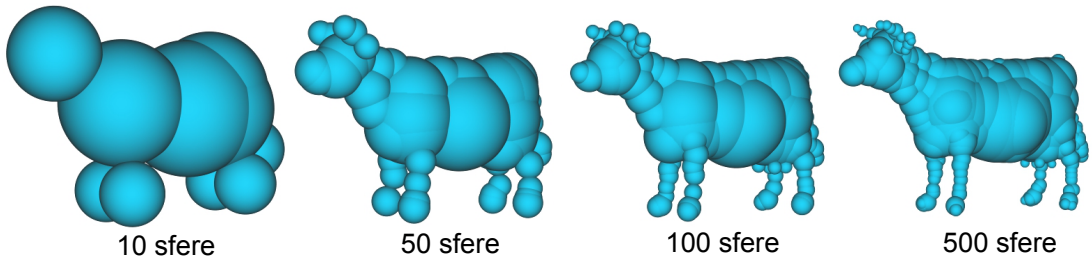
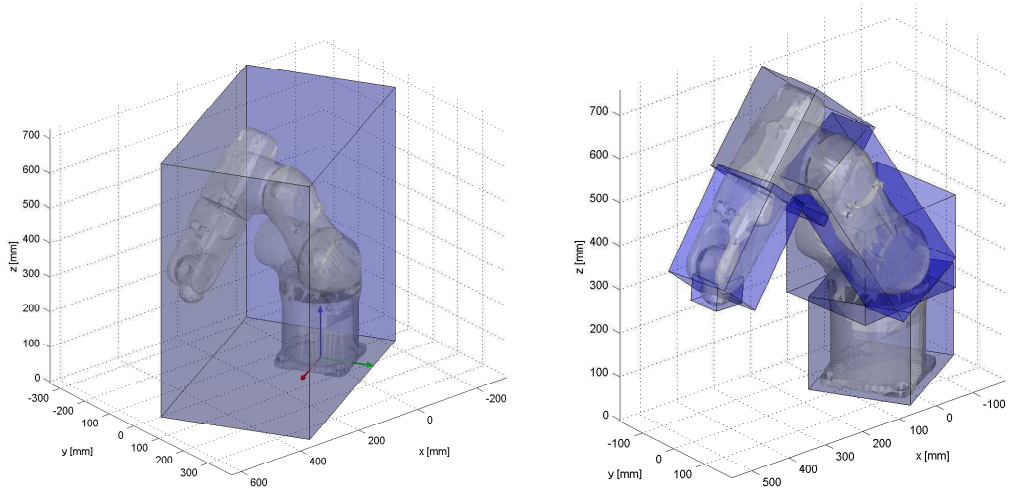


Figura 1.6: Esempio di algoritmo automatizzato per BVH utilizzando sfere [4].

In figura 1.6 è mostrato un esempio di BVH utilizzando sfere per rappresentare un oggetto. L'applicazione nella robotica potrebbe essere quella di rappresentare dapprima l'intero robot e successivamente verificare il test di collisione per ogni link (figura 1.7). In questo caso il primo box oltre ad avere un volume iniziale eccessivamente grande, è necessario doverlo aggiornare ad ogni variazione



(a) Prima verifica di collisione.

(b) Seconda verifica per tutti i link.

Figura 1.7: Possibile applicazione di BVH per manipolatore robotico.

della configurazione del robot. Conviene quindi valutare direttamente il test di collisione per ogni link, evitando un passaggio iniziale sostanzialmente inutile.

1.2 Cenni ad altre tecniche di collision detection

Diversamente dal bounding volume (e BVH) che possono adattarsi ragionevolmente bene a qualsiasi tipo di applicazione, esistono altri approcci al collison detection che possono essere applicati in solo alcuni contesti. Queste tecniche comprendono tre diverse categorie:

- Spatial partitioning
- BSP tree hierarchy
- Convexity-based

1.2.1 Spatial partitioning

Le tecniche di partizionamento dello spazio sono utilizzate per broad phase operando una suddivisione dello spazio in regioni, e verificare se gli oggetti intersecano la stessa regione dello spazio. Dal momento che gli oggetti potrebbero collidere

solo se occupano la stessa regione, il numero di test pair-wise è drasticamente ridotto.

Confrontando schemi BVH con spatial partitioning, la differenza principale è che nel primo caso gli oggetti sono inseriti in un singolo volume, mentre nel secondo caso gli oggetti sono tipicamente rappresentati in più celle o partizioni.

Esistono diverse tipologie di spatial partitioning fra cui:

- Griglia di dimensione uniforme
- Rappresentazione gerarchica: octree e k-d tree
- Proiezione sugli assi

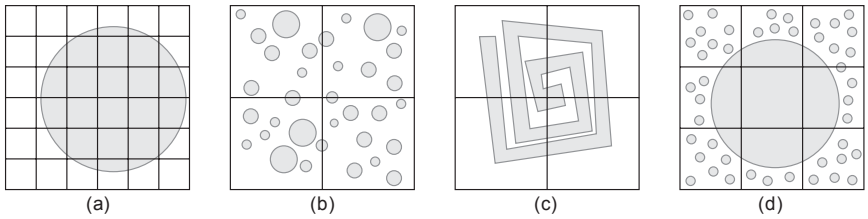
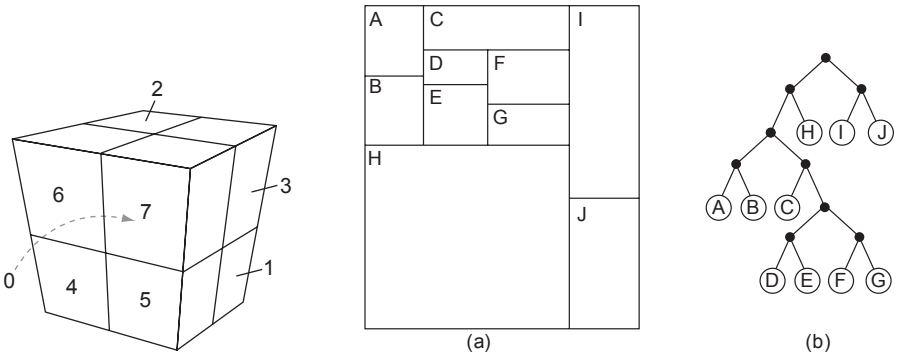


Figura 1.8: Problema relativo alla dimensione delle celle uniformi nel 2D. (a) fine (b) grossolano rispetto agli oggetti (c) grossolano rispetto alla complessità dell'oggetto (d) nè fine nè grossolano.

La suddivisione dello spazio in griglie regolari, di egual dimensione, incorre nel problema di determinare, al fine delle performance, le dimensioni delle celle (vedi figura 1.8). Altri sottoproblemi riguardano la rappresentazione della struttura (matrici od array) e gli algoritmi di ricerca.

Attraverso una rappresentazione gerarchica dello spazio si ottengono prestazioni migliori. Il principio è quello di suddividere iterativamente le celle in volumi sempre più piccoli. La rappresentazione che viene utilizzata riguarda solitamente una struttura ad albero. Se ad ogni iterazione si suddivide lo spazio 3D in otto sezioni uguali si ottiene un octree (figura 1.9a), se la suddivisione non è omogenea si parla invece di k-d tree (figura 1.9b).

Il test broad-phase può essere realizzato anche mediante la proiezione dei limiti degli oggetti su un asse e la verifica di collisione involve nell'intersezione di intervalli 1D. Il principio di funzionamento è mostrato in figura 1.10.



(a) Octree: la radice possiede otto nodi foglia.

(b) Un k-d tree nel 2D: (a) decomposizione dello spazio e (b) rappresentazione ad albero.

Figura 1.9: Rappresentazione gerarchica ad albero.

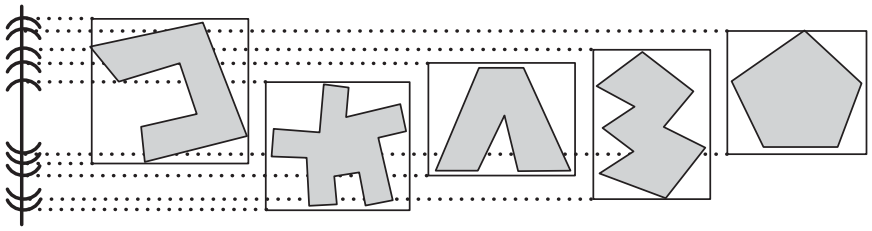


Figura 1.10: La proiezione degli oggetti sull'asse y comporta esito sbagliato di collisione a causa dell'intersezione degli intervalli.

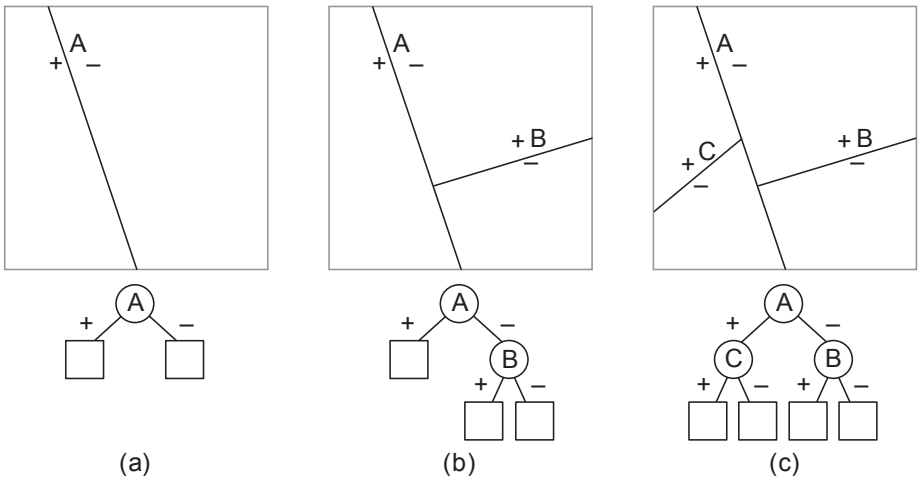
1.2.2 BSP Tree Hierarchies

Una generalizzazione delle tecniche di spatial partitioning è la rappresentazione a partizione binaria (BSP tree, binary space-partitioning tree).

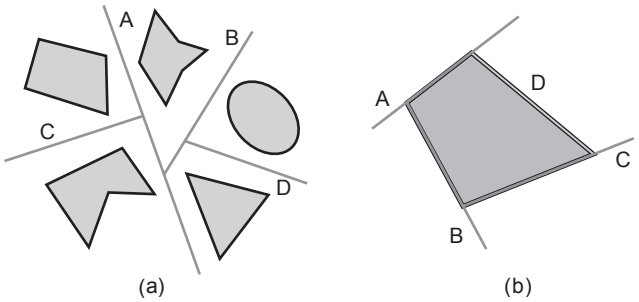
Essa consiste in una struttura ad albero binario che partiziona ricorsivamente lo spazio in una coppia di sottospazi rispetto ad un piano di posizione e orientazione arbitraria. Un esempio di principio di funzionamento è mostrato in figura 1.11a.

Nell'ambito del collision detection un BSP tree è molto versatile in quanto può costituire sia come spatial partitioning sia come la rappresentazione del volume di un oggetto (figura 1.11b).

Un BSP tree viene usato principalmente per rappresentare la geometria statica dell'ambiente.



(a) Struttura ad albero binario per: (a) split iniziale (b) primo split di secondo livello (c) secondo split di secondo livello



(b) Utilizzo come (a) partizione dello spazio (b) rappresentazione del bounding volume

Figura 1.11: Rappresentazione a partizione binaria.

1.2.3 Convexity-based

Oggetti convessi possiedono certe proprietà che li rendono particolarmente adatti per collision detection. Non è un caso che tutti i bounding volume esistenti siano oggetti convessi.

Le due proprietà più importanti sono:

1. Esistenza di piani di separazione che non intersecano con oggetti convessi.

2. La distanza fra due punti di due oggetti convessi è sempre un minimo locale (figura 1.12).

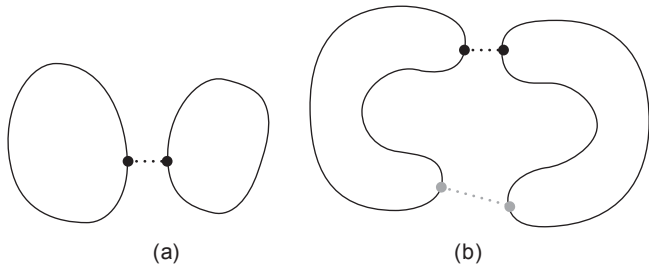


Figura 1.12: (a) per due oggetti la distanza minima è sempre un minimo globale (b) per due oggetti convessi la distanza minima non è necessariamente un minimo globale

La prima proprietà oltre ad essere utilizzata come prova di collisione, nel caso di poliedri convessi essa consente suddividere lo spazio in due regioni distinte: interno ed esterno.

La seconda caratteristica può essere sfruttata per migliorare i casi in cui il movimento o rotazione fra due oggetti avviene di una piccola quantità, cosicché è possibile prevedere quali features (vertici, lati, facce) sono più probabili alla collisione.

Capitolo 2

Rappresentazione del modello

La prima operazione da compiere per realizzare un sistema di collision detection riguarda, come detto nel capitolo 1, la scelta della rappresentazione del modello degli oggetti.
Nell'ambito della robotica i corpi di riferimento da descrivere tramite un modello sono essenzialmente due:

- **Robot**: manipolatore meccanico che costituisce la componente dinamica del sistema tramite il movimento di traslazione e/o rotazione rigida dei propri link.

- **Ostacoli**: ingombri degli oggetti presenti nello spazio di lavoro del robot e nella cella robotizzata, considerati come oggetti statici.

La scelta fatta riguarda la definizione tridimensionale di queste due componenti attraverso due distinte modalità di rappresentazione: *modello geometrico* o *insieme di punti*.

Nel primo caso si tratta di rappresentare il modello del robot o dell'ostacolo mediante il più appropriato bounding volume, secondo i criteri descritti nel capitolo precedente.

L'utilizzo di un insieme di punti appartenenti alla superficie dell'oggetto può essere considerata la forma più semplice di rappresentazione. Questa scelta consente di descrivere semplicemente la forma di un oggetto geometrico anche complesso, come può essere un robot.

Il problema di collision detection si traduce quindi nello studiare le iterazioni che sussistono fra robot e ostacoli descritti da modelli che possono essere arbitrariamente modelli geometrici e/o insieme di punti nello spazio.

In questo capitolo si vuole descrivere la strategia utilizzata per la rappresentazione del modello di robot attraverso un'insieme di punti.

2.0.4 Generazione dei link in formato STL

STL, acronimo di *Standard Triangulation Language*, è un formato di file, binario o ASCII, utilizzato nella prototipazione rapida attraverso software CAD. Un file .stl rappresenta un solido la cui superficie è stata discretizzata in triangoli. Esso consiste delle coordinate x, y e z ripetute per ciascuno dei tre vertici di ciascun triangolo, con un vettore per descrivere l'orientazione della normale alla superficie.

Il formato STL presenta dei vantaggi quali la semplicità, in quanto risulta molto facile da generare mediante software di gestione grafica (come SolidWork) e la possibilità di processare la struttura dati tramite MATLAB al fine di generare la nuvola di punti e implementare la simulazione grafica.

In pratica è necessario importare il modello CAD del robot, disponibile in ogni sito del costruttore, tramite SolidWork (o qualsiasi altro software di gestione grafica) e formattarlo come file .stl per poter essere importato da MATLAB. Questo passo è essenziale in quanto fra i modelli CAD del robot disponibili in rete non è presente nativamente il formato STL, bensì formati standard quali: Parasolid, SolidWork, IGES, STEP, ecc.

Qualunque siano i gradi di libertà del robot adottato (le possibilità come detto sono 4 o 6 gdl), ciò che viene fornito dal sito del costruttore sono (devono essere) i modelli CAD dei singoli link del robot. Questa rappresenta la condizione necessaria e sufficiente a poter utilizzare correttamente l'interfaccia grafica e definire la catena cinematica di Denavith Hartemberg del robot.

Le semplici operazioni preliminari da svolgere per mezzo di un software 3D devono essere svolte in modo sistematico. Esse sono:

1. Download del modello CAD del robot in un qualsiasi formato.

2. Importazione di un link per volta nel software di gestione grafica.

3. Definizione di un sistema di coordinate per ogni link compatibile con la catena cinematica di Denavith-Hartemberg (figura)

4. Impostare le opzioni di esportazione STL come in figura.

5. Salvare i link come “link#.stl”, dove # indica numero progressivo crescente a partire da 0 della base robot, all’interno di una stessa folder nominata con una sigla identificativa del robot.

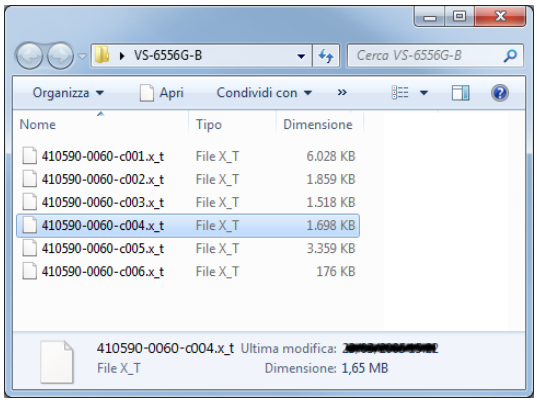


Figura 2.1: Esempio di cartella contenente i file nel formato 3-D Parasolid di un robot a 6 gradi di libertà.

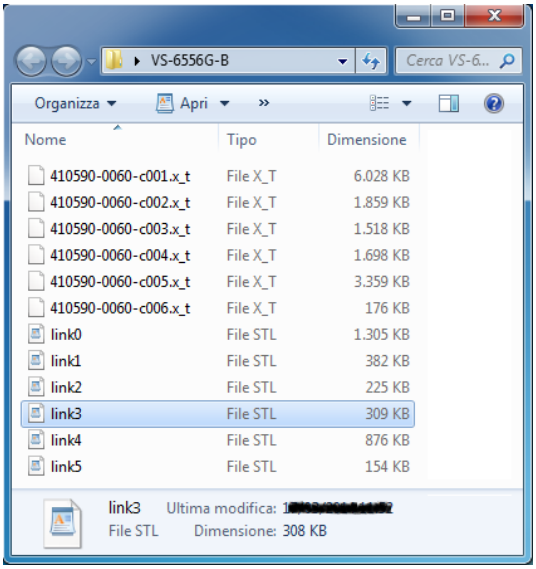
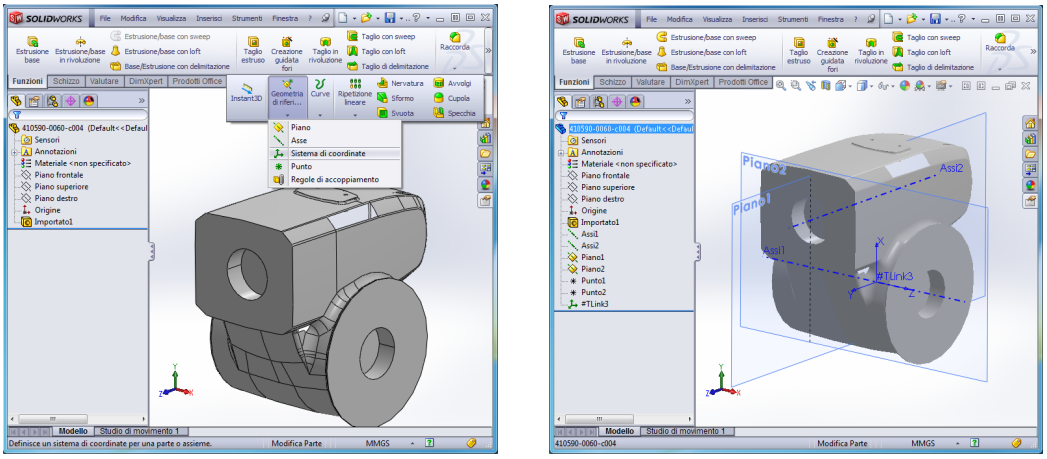


Figura 2.2: Esempio di cartella con dati STL rinominati ed elaborati con software grafico.

Se il formato scelto per il download è 3-D Parasolid, i file nativi saranno del tipo mostrato in figura 2.1. Il modello CAD scelto come esempio è un robot a 6 gradi di libertà di marca DENSO con sigla identificativa VS-6556B-G. Alla fine delle operazioni descritte sopra, il risultato finale dovrà essere una cartella contenente i file STL dei link come mostrato in figura 2.2.

La figura 2.3a mostra la grafica iniziale a seguito dell’importazione tramite SolidWork del file Parasolid corrispondente al link 3, e la definizione di un sistema di coordinate compatibile con Denavith-Hartemberg (figura 2.3b). Questa operazione richiede una minima dimestichezza con il software grafico; nel caso



(a) Grafica iniziale. (b) Definizione della terna di DH.

Figura 2.3: Esempio di importazione di un link mediate SolidWork.

del link 3 si è reso necessario definire anche due piani (eventualmente si possono utilizzare il piano frontale, superiore e destro di default), due assi e due punti.

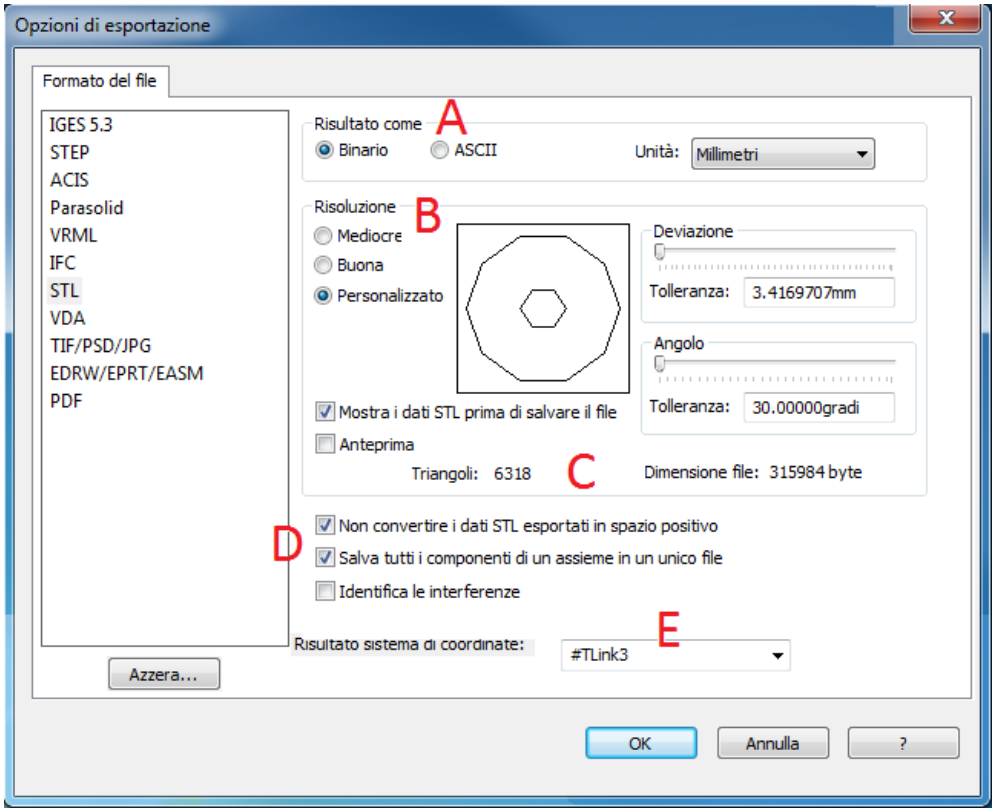


Figura 2.4: Opzioni di esportazione in formato STL.

Per convertire in formato STL è sufficiente salvare e rinominare il file. Le opzioni di esportazione devono essere quelle mostrate in figura 2.4. In particolare, in (A) si seleziona il formato binario e l'unità di misura in millimetri. La risoluzione in (B) è sufficiente sia la minima possibile (equivalente all'impostazione mediocre). L'anteprima (C) mostra il numero di triangoli e le dimensioni del file. Il primo checkbox in (D) è essenziale per esportare il file con i triangoli definiti rispetto al sistema di coordinate selezionato dal menù a tendina (E). Il secondo checkbox in (D) è utilizzato nel caso in cui il modello CAD di un link presenti più parti da riferire alla stesso sistema di coordinate. Ovviamente in (E) è essenziale selezionare la terna DH del link corrispondente definita precedentemente dall'utente per mezzo di SolidWord.

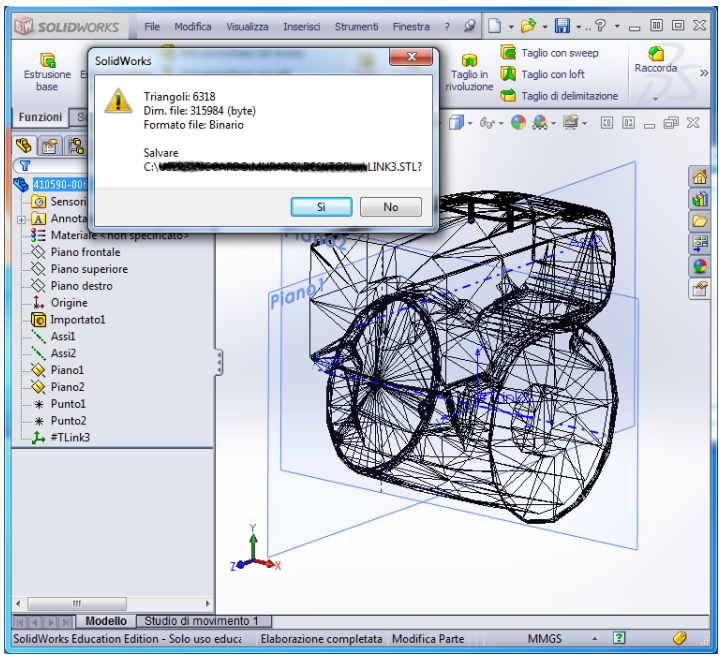


Figura 2.5: Anteprima di visualizzazione della triangolazione del link.

Un esempio di anteprima di triangolazione STL successiva al salvataggio è mostrata in figura 2.5. Si può osservare come i triangoli presentino diverse dimensioni; risultano infatti particolarmente stretti a allungati lungo bordo e zone di maggior curvatura nella geometria del link. Questo è essenzialmente la causa della minima risoluzione di esportazione impostata.

2.1 Guide User Interface

Utilizzando MATLAB è stato progettato ed implementato un'interfaccia utente che permette di importare ed elaborare un qualsiasi modello CAD di un robot a 6 o 4 gdl in formato STL.

Le operazioni che è possibile fare con questa GUI saranno analizzate nelle prossime sezioni e riguardano:

1. importazione del modello di un robot
2. impostare dei parametri per la visualizzazione tridimensionale
3. impostare dei parametri per la generazione di punti per collision detection
4. impostare i parametri per tabella di Denavith Hartemberg
5. definizione di bounding volume di ogni link per collision detection
6. salvataggio dei dati compatibili con il simulatore

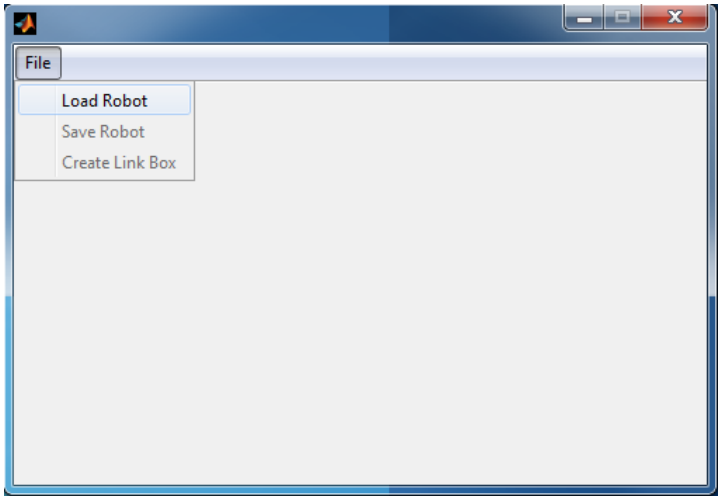


Figura 2.6: Anteprima di visualizzazione iniziale della GUI.

Inizialmente la GUI si presenta come in figura 2.6, ed è consentito solo importare il modello di un robot (selezione Load Robot).

2.1.1 Importazione del modello di un robot

Il modello CAD dei link del robot dovranno essere elaborati come descritto nella sezione 2.0.4 al fine di ricavare una cartella contenente i dati dei link in formato STL come mostrato nell'esempio di figura 2.2. Pertanto è essenziale che i dati siano nominati come "link#.stl" con # ad indicare il numero progressivo crescente a partire da 0 della base robot.

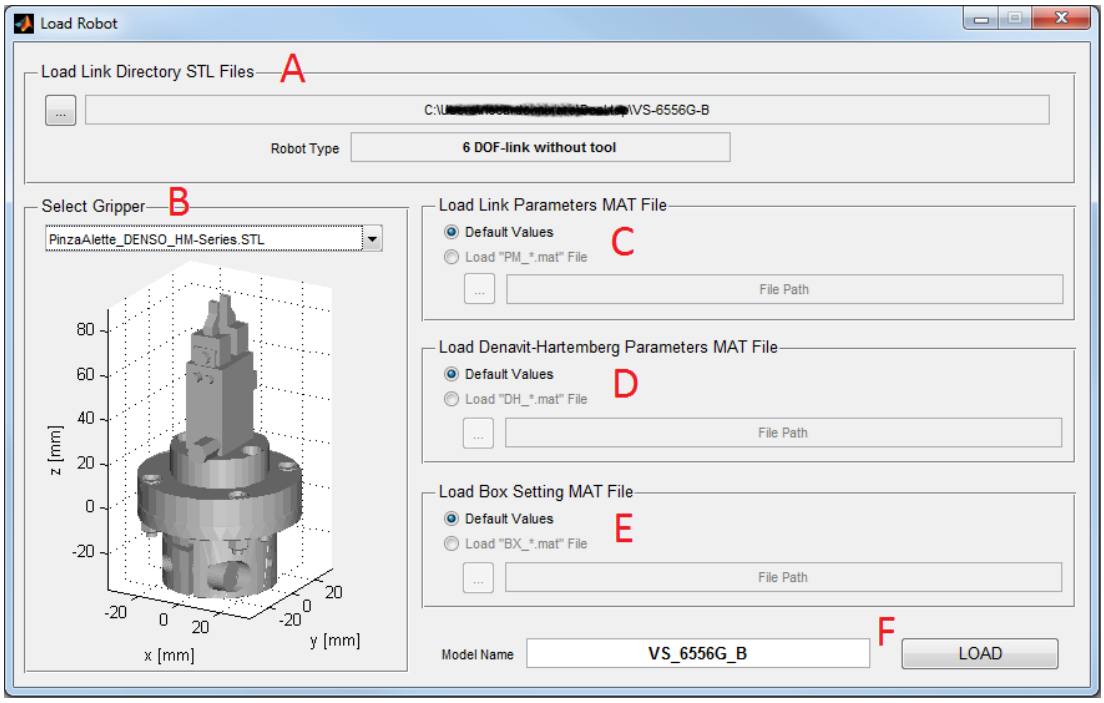


Figura 2.7: Finestra per importazione del modello del robot.

La finestra per il caricamento dei dati è mostrata in figura 2.7. Nel pannello (A) è possibile selezionare la directory path contenente i file dei link del robot in formato STL. Viene evidenziato anche la tipologia di robot selezionato; in questo caso 6 gradi di libertà senza utensile (tool). Le altre possibilità riguardano robot a 4 gradi di libertà con o senza utensile. Nel caso in cui i dati non siano corretti, viene visualizzato un messaggio di errore (figura 2.8) e naturalmente non è possibile caricare i dati.

Il programma impone la scelta di un utensile fra una serie di gripper di default (B) da utilizzare come end-effector del robot. In questo caso, non essendo presente il tool, verrà generato il file link#.stl del tool selezionato all'interno della cartella

(in questo caso link6.stl). Se già presente comparirà come scelta di default. É infatti sempre possibile definire un utensile che verrà gestito dal programma come fosse un normale link del robot.

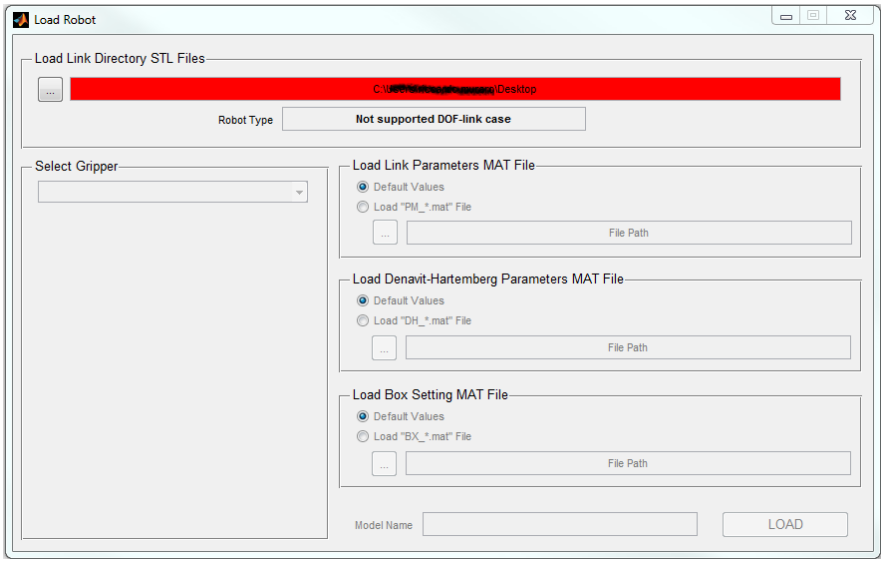


Figura 2.8: Esempio di errore di importazione del modello del robot.

I pannelli (C), (D) e (E) riguardano l'importazione di determinati file in formato .mat contenenti valori dei parametri precedentemente salvati. Alla prima creazione del modello tali parametri saranno impostati con valori di default. Tuttavia è sempre possibile selezionare i parametri di un altro modello di robot purché compatibile con il numero di gradi di libertà, indipendentemente che siano corretti o meno. La gestione di errori in caso di file non compatibili con il robot avviene equivalentemente al caso (A) precedente.

Il significato di questo parametri risulterà più chiaro con il proseguo della trattazione quando ne verrà spiegata la loro funzione; in ogni caso questi file riguardano:

- (C) Visualizzazione tridimensionale e generazione dei punti per il collision detection;
- (D) Tabella di Denavith Hartemberg;
- (E) Settaggio dei bounding volume definiti mediante box.

In fondo alla finestra di caricamento (F) è eventualmente possibile rinominare il nome/sigla del robot che di default corrisponde al nome della cartella selezionata. Tuttavia sono consentiti come caratteri solo lettere maiuscole, numeri e _ (underscore), tutte le altre possibilità sono escluse. Il pulsante "LOAD" termina la fase di importazione del modello e rimanda l'utente alla GUI principale.

2.1.2 Finestra principale

Effettuato il caricamento del modello robot, compare l'interfaccia principale mostrata in figura 2.9. Essa è composta da 3 sezioni :

- (A) Opzioni di visualizzazione
- (B) Anteprima di visualizzazione
- (C) Impostazione dei parametri

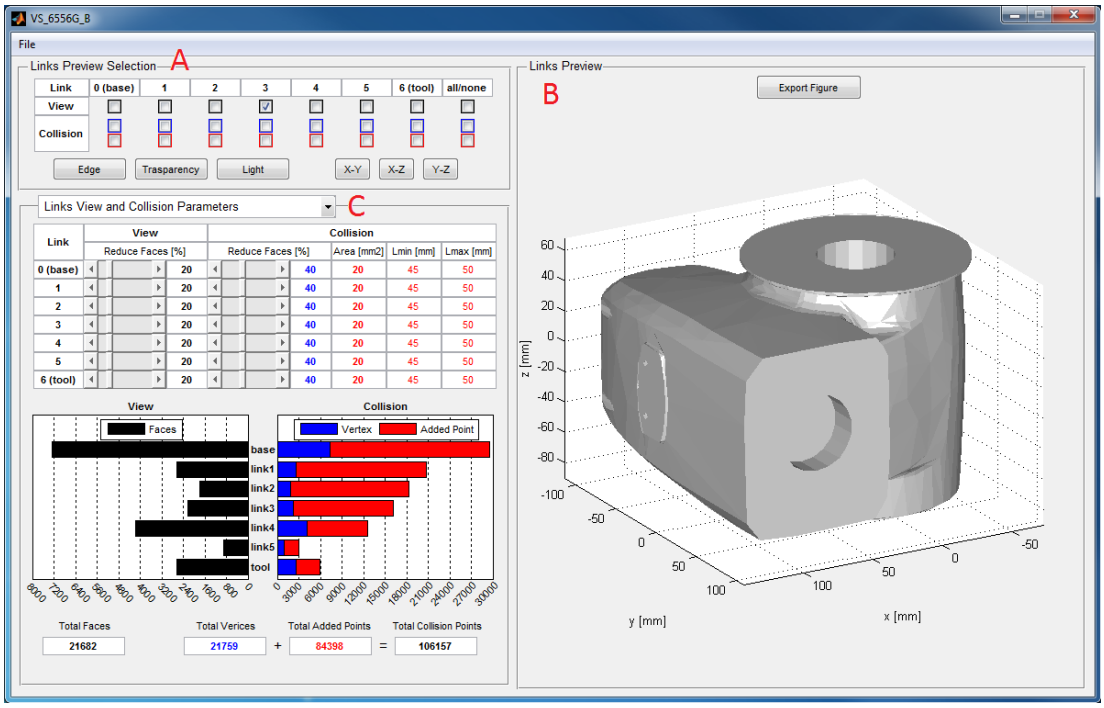


Figura 2.9: Finestra principale della GUI.

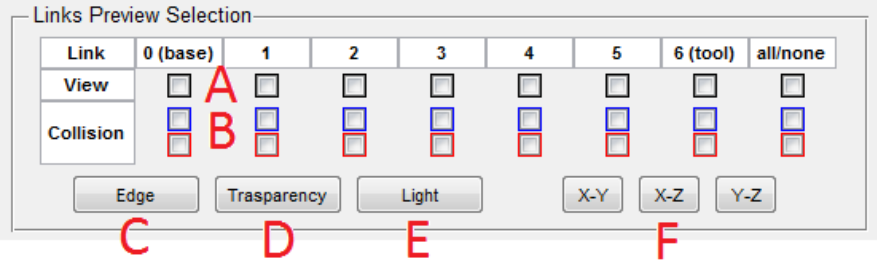


Figura 2.10: Pannello per le opzioni di visualizzazione.

2.1.3 Opzioni di visualizzazione

Tramite questo pannello è possibile gestire l'anteprima di visualizzazione dei link. Esso è quindi legato con il pannello "Links Preview". Con riferimento alla figura 2.10, le funzionalità riguardano:

- (A) selezione dei link da visualizzare (checkbox nero)
- (B) visualizzazione dei punti per collision detection
  - punti dei vertici dei triangoli (checkbox blu)
  - punti di raffinamento dei triangoli (checkbox rosso)
- (C) visualizzazione dei lati (edge) dei triangoli
- (D) trasparenza dei link e visualizzazione del frame
- (E) impostazione della luce prospettica
- (F) visualizzazione nei tre piani XY, XZ e YZ

Selezione dei link

Il numero di colonne presenti corrisponde al numero di link da selezionare (pari ai gradi di libertà), con l'aggiunta di un campo finale "all/none" per selezionare tutti i checkbox della riga corrispondente. É possibile quindi una selezione multipla. Per ogni link la selezione riguarda tre campi: il primo per la visualizzazione, gli altri due per aggiungere la visualizzazione dei punti.

Nella figura 2.11 è presente una panoramica dei link del robot utilizzato come esempio. Se si selezionano tutti i checkbox della prima riga senza impostare i

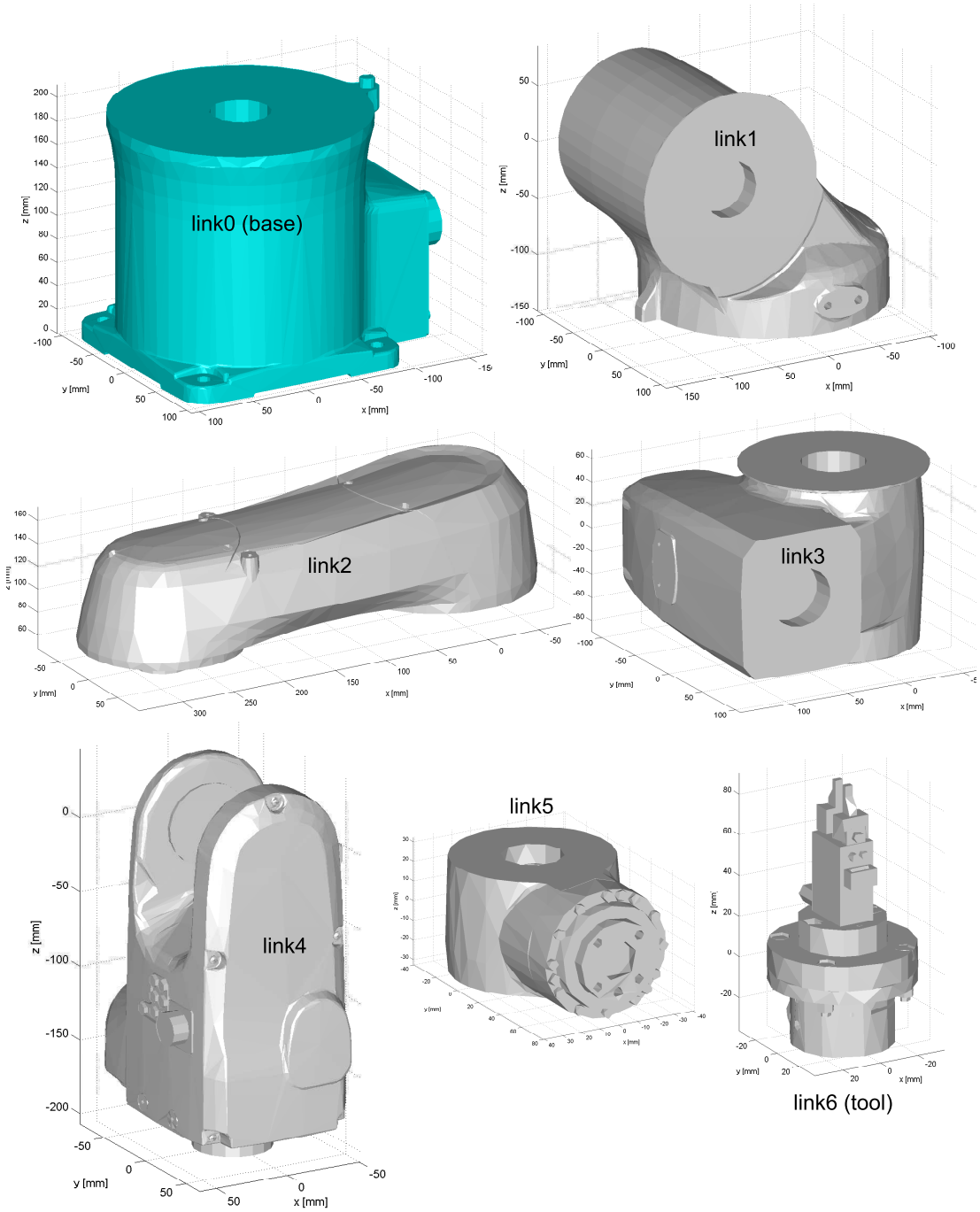


Figura 2.11: Visualizzazione di ogni link del robot DENSO VS-6556G-B.

parametri di Denavith Hartemberg (vedi sezione .....), il risultato è quello mostrato figura 2.12. Si rimanda alla sezione ..... per chiarire il significato annesso alle selezioni per il collision.

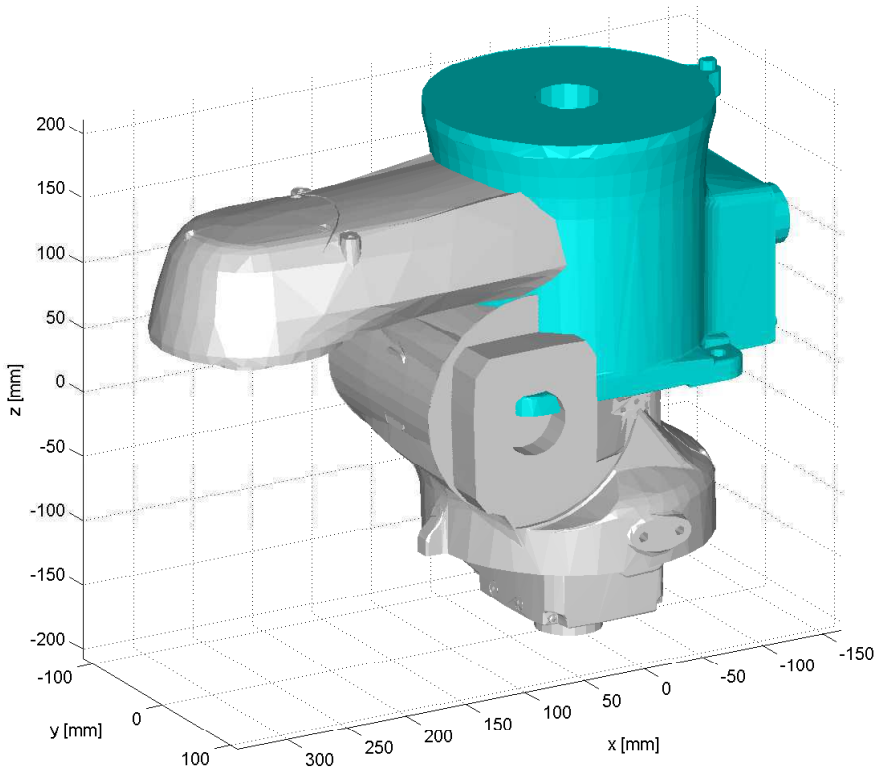


Figura 2.12: Selezione multipla di tutti i link senza impostazione dei parametri di Denavith Hartemberg.

Effetto dei pulsanti

I pulsanti “Edge” e “Trasparency” assumono il carattere di on/off (togglebutton) con effetto su tutti i link selezionati per la visualizzazione. Quando è attivato il pulsante “Edge” vengono visualizzati i lati dei triangoli (figura 2.13). Attivando invece il pulsante “Trasparency” viene mostrato il sistema di riferimento (frame) dei link selezionati (figura 2.14). Essendo il frame situato generalmente all’interno alla geometria del link, quest’ultimo diviene semi-trasparente.

Se entrambi questi togglebotton sono attivi, la trasparenza ha la priorità rispetto gli edge. Tuttavia, per i link nei quali siano visualizzati i punti, la trasparenza non ha effetto.

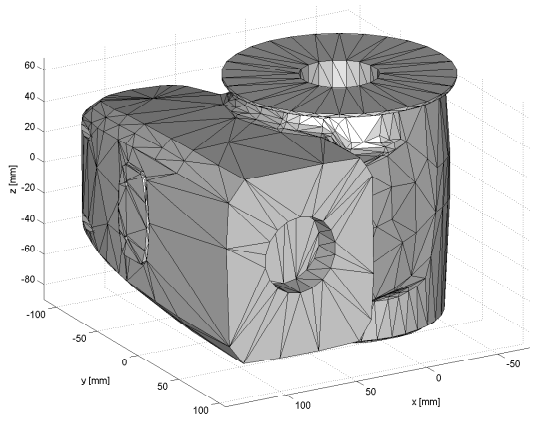


Figura 2.13: Effetto del pulsante Edge.

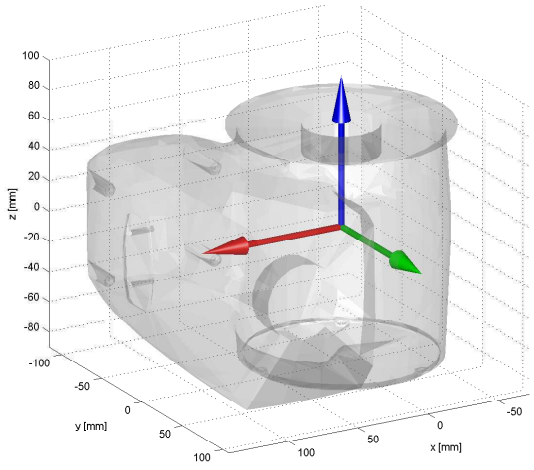
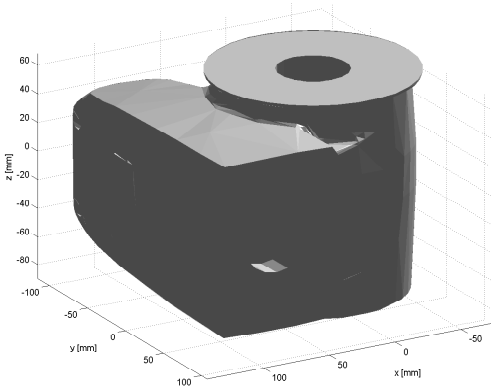
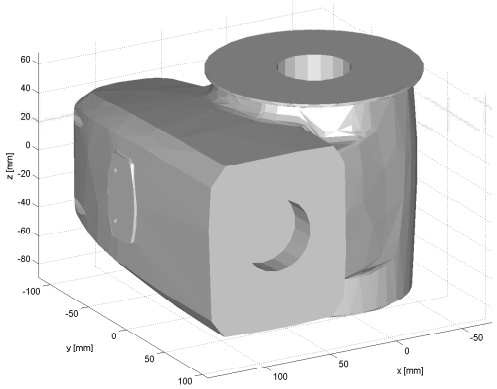


Figura 2.14: Effetto del pulsante Trasparency.



(a) Situazione precedente all'illuminazione.

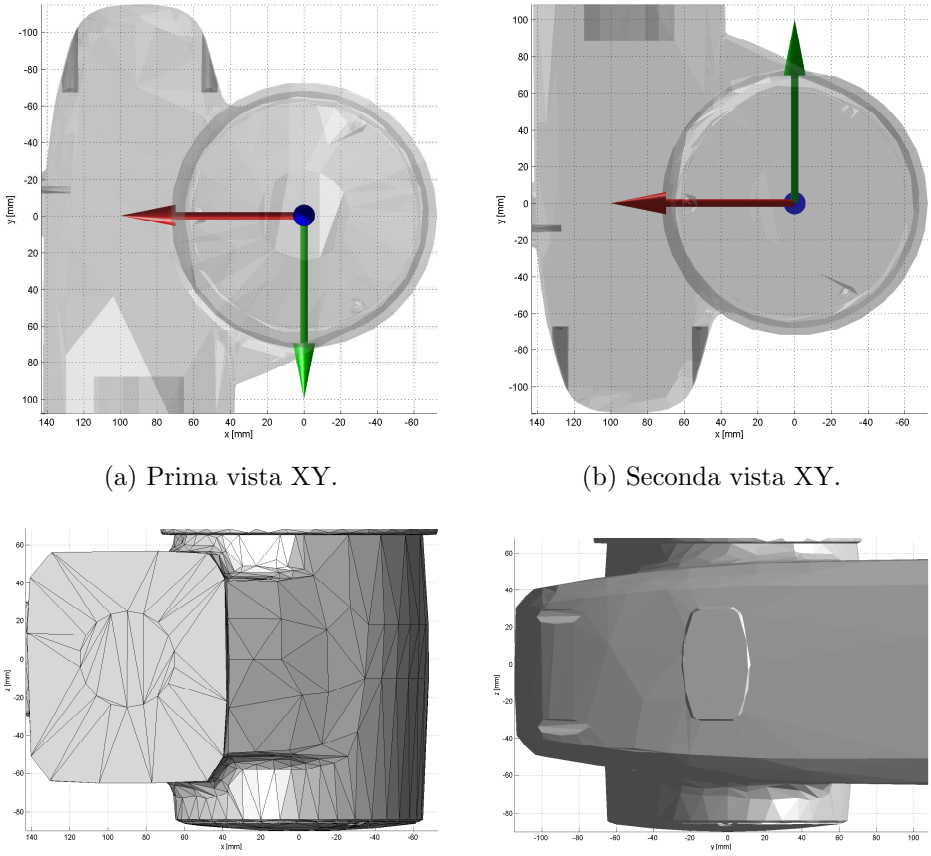


(b) Situazione successiva all'illuminazione.

Figura 2.15: Effetto del pulsante Light.

Per agevolare l'utente nella visualizzazione, è presente un pulsante "Light" per illuminare le parti che risulterebbero oscurate. L'effetto è quello mostrato in figura 2.15.

Vi sono infine 3 pulsanti per le viste prospettiche "XY","XZ" e "YZ"(figura 2.16). Premendo due volte successive uno di questi bottoni viene invertita la vista del terzo versore passando da uscente ad entrante (figura 2.16a e 2.16b).



(a) Prima vista XY.

(b) Seconda vista XY.

(c) Vista XZ con edge.

(d) Vista YZ.

Figura 2.16: Effetto delle viste prospettiche.

2.1.4 Anteprima di visualizzazione

La sezione destra della GUI è dedicata all'anteprima di visualizzazione. Ci sono due funzionalità:

- Esportazione della figura, mediante pulsante "Export Figure", per consentire all'utente lo zoom di particolari, pan, salvataggio del grafico ecc.
- Ripristino della vista di default mediante doppio clic del mouse.

In fase di progettazione della GUI non è stato possibile implementare lo zoom del grafico; il pulsante per l'esportazione del plot ha risolto in parte questo inconveniente. La funzionalità di rotazione dell'immagine invece è sempre attiva ed è sempre possibile ruotare il grafico come si desidera.

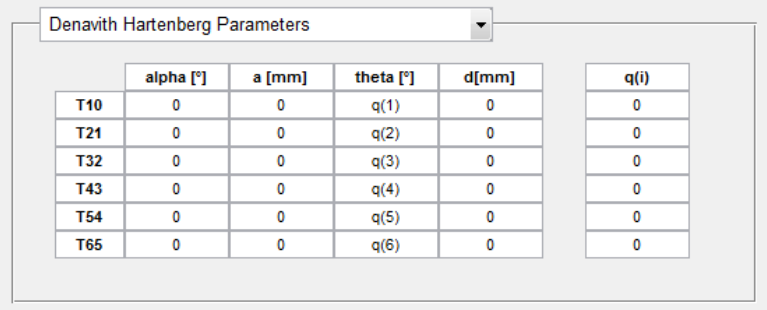
2.1.5 Pannello per impostazione dei parametri

Dalla finestra principale, mediante un menù a tendina (vedi (C) di figura 2.9 a pagina 27) è possibile selezionare ed impostare due categorie indipendenti di parametri. Esse riguardano:

- parametri per visualizzazione e raffinamento punti dei link ("Link view and collision parameters")
- parametri di Denavith-Hartenberg ("Denavith-Hartenberg parameters")

2.2 Impostazione parametri di Denavith-Hartenberg

Ad ogni link (esclusa la base riferita all'origine) sono associati i quattro parametri di Denavith Hartemberg: α, a, $theta$ e d. Si rimanda all'appendice B o ad un qualsiasi testo di robotica per comprenderne il loro significato.



	alpha [°]	a [mm]	theta [°]	d[mm]	q(i)
T10	0	0	q(1)	0	0
T21	0	0	q(2)	0	0
T32	0	0	q(3)	0	0
T43	0	0	q(4)	0	0
T54	0	0	q(5)	0	0
T65	0	0	q(6)	0	0

Figura 2.17: Tabella di Denavith Hartemberg con valori di default per robot a 6 gradi di libertà.

La tabella presente nell'interfaccia grafica, mostrata in figura 2.17, inizialmente è impostata con dei valori di default. All'utente il doppio compito di:

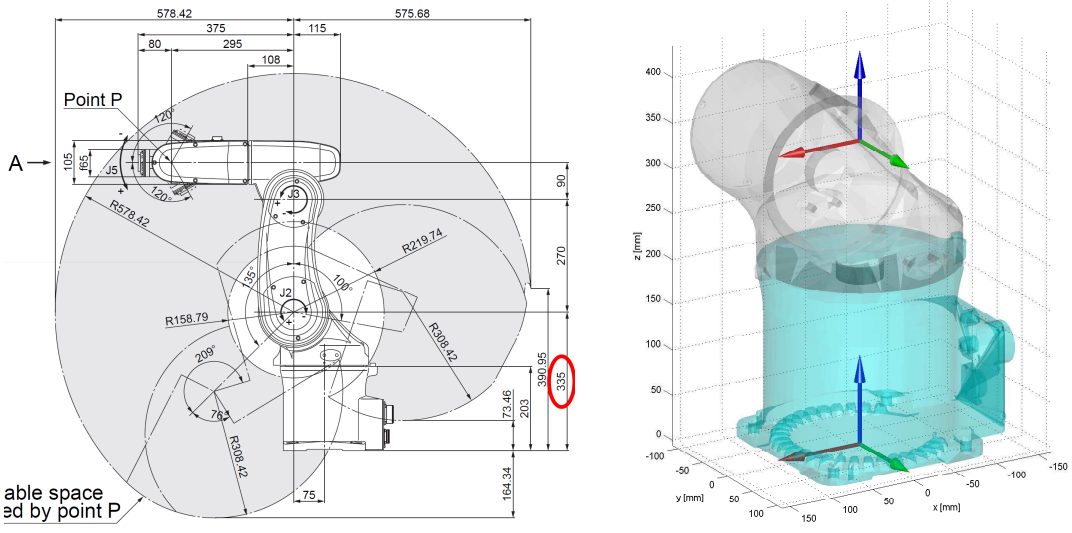
1. inserire i parametri dimensionali
2. determinare i parametri responsabili del movimento per ogni link

Nel primo caso è sufficiente consultare le specifiche dimensionali del robot presenti nel manuale od eventualmente aiutarsi con delle quote dimensionali impartite tramite SolidWork in fase di importazione dei link in formato STL.

I parametri responsabili del movimento corrispondono ai gradi di libertà impartiti dai motore e vengono determinati nella tabella dalle variabili $q(i)$, dove i corrisponde al numero di riga. Se il parametro consiste in una rotazione, la variabile dovrà essere presente nella colonna θ; se consiste in una traslazione, tale variabile sarà impartita nella colonna d. Secondo DH, i parametri α e a non possono mai essere parametri variabili.

É possibile quindi inserire nei campi dalla tabella delle espressioni nelle quali l'unica variabile consentita è $q(i)$. In caso di un qualsiasi errore di inserimento in un campo della tabella, l'interfaccia evidenzia istantaneamente in rosso la casella corrispondente.

A riprova del fatto che i parametri responsabili del movimento sono dei valori variabili, a destra della tabella di DH è presente una colonna "q(i)" nella quale è possibile impostare il corrispondente valore numerico ed eventualmente testare la correttezza dei valori di DH inseriti. Ovviamente l'unità di misura sarà ° (gradi) se la variabile corrispondente è una rotazione, mm (millimetri) se consiste in una traslazione.



(a) Estratto delle specifiche dimensionali. (b) Risultato impostazione parametro d.

Figura 2.18: Esempio completamento riga T10 della tabella.

A titolo di esempio si vuole completare la prima riga della tabella, ovvero impostare la corrispondenza fra la base e il link 1. Essa consiste nella sola tra-

slazione lungo l'asse Z, quindi è sufficiente in questo caso ricavare il parametro d. Dalle specifiche dimensionali del robot in questione (figura 2.18a) si evince che tale valore dev'essere 335 $[mm]$ e l'effetto di tale impostazione sul link 1 è mostrato in figura 2.18b.

Denavith Hartenberg Parameters

	alpha [°]	a [mm]	theta [°]	d[mm]	q(i)
T10	0	0	q(1)	335	0
T21	-90	75	q(2)-90	0	0
T32	0	270	q(3)	0	0
T43	-90	90	q(4)	295	0
T54	-90	0	180-q(5)	0	0
T65	-90	0	180+q(6)	80	0

Figura 2.19: Tabella di Denavith Hartemberg definita completamente.

Procedendo similmente si completa la tabella di DH, mostrata in figura 2.19, giungendo alla definizione completa del robot (figura 2.20). In questo caso le 6 coppie cinematiche del robot sono tutte giunti rotoidali, e le loro rotazioni sono state rese conformi alle specifiche del robot tramite l'aggiunta un angolo statico ed eventualmente un segno meno per conferire il verso di rotazione corretto.

2.3 Link view and collision parameters

Con riferimento al pannello mostrato in figura 2.21, le componenti di cui è composto sono:

- (A) Impostazione risoluzione triangoli per visualizzazione
- (B) Impostazione risoluzione vertici dei triangoli
- (C) Impostazione raffinamento punti interni ai triangoli
- (D) Istogramma per panoramica impostazioni in (A)
- (E) Istogramma per panoramica impostazioni in (B) e (C)
- (F) Numero totale di triangoli
- (G) Numero totale di punti per test collisione

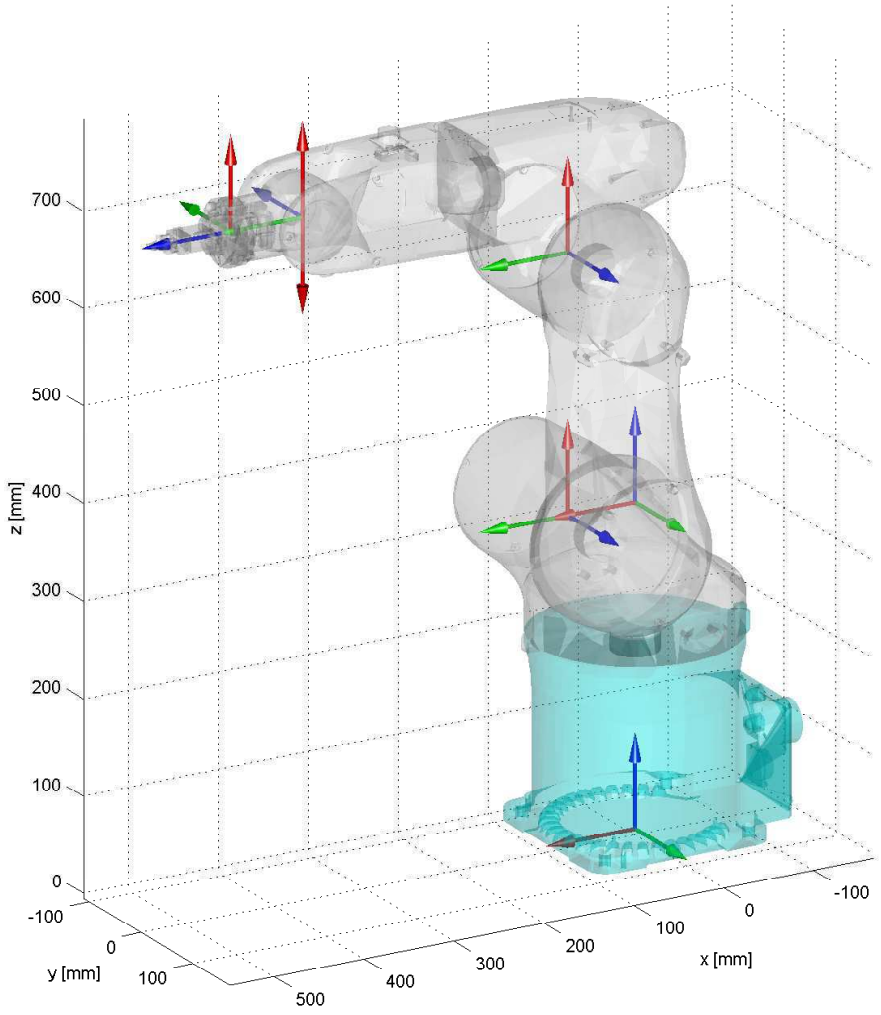


Figura 2.20: Visualizzazione del robot con impostazione dei paramatri di DH.

2.3.1 Risoluzione della visualizzazione

Tramite degli slider è possibile impostare la percentuale di risoluzione per la visualizzazione dei singoli link. Tale valore dev'essere compreso fra un minimo del 5 % fino al 100 % dato dalla risoluzione massima.

La variazione del numero di triangoli avviene proporzionalmente alla percentuale di risoluzione anche se con un coefficiente angolare diverso per ogni link, come raffigurato nel grafico di figura 2.22.

L'effetto impartito sul link 4 per diversi valori di risoluzione è mostrato in figura 2.23. Si può osservare che utilizzando una risoluzione elevata non comporti un miglioramento di visualizzazione tale da giustificare l'aumento del numero di

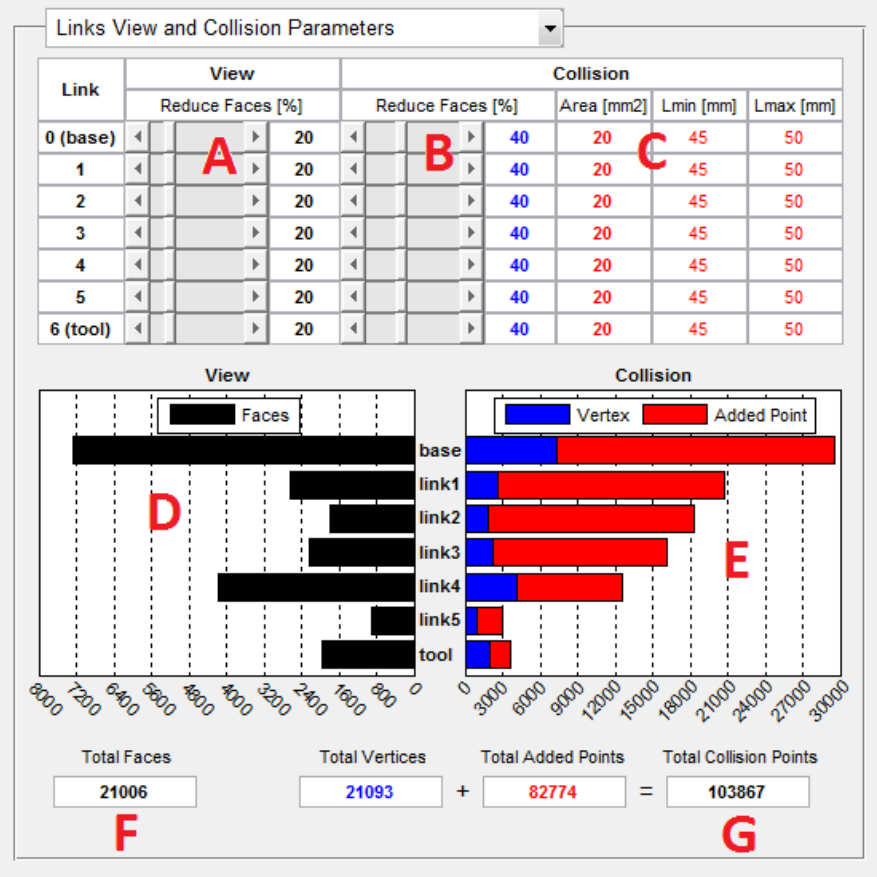


Figura 2.21: Finestra della GUI per l'impostazione dei parametri.

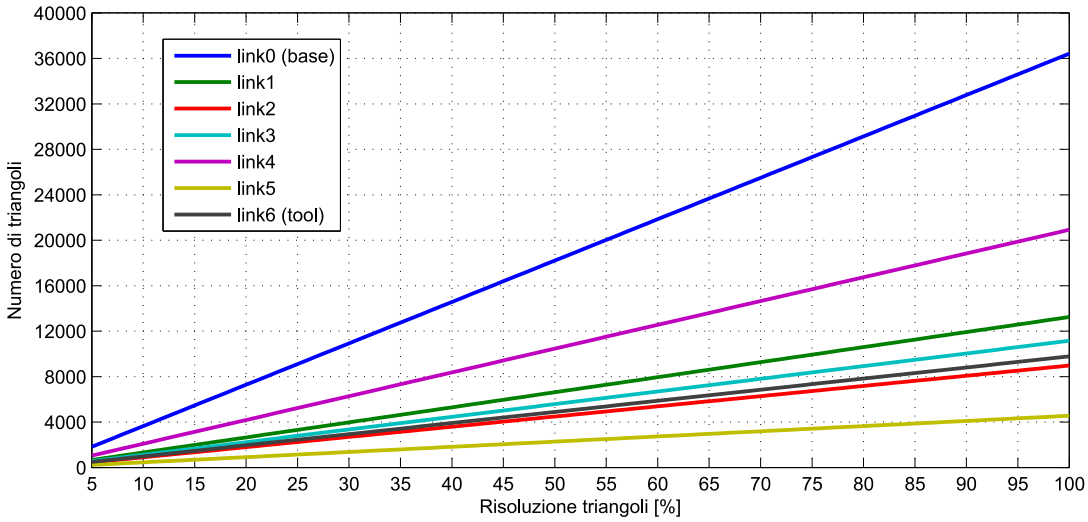


Figura 2.22: Variazione del numero di triangoli in funzione della risoluzione di visualizzazione per i link.

triangoli.

Il significato di queste impostazioni risiede nella velocità di MATLAB nel processare e raffigurare il movimento del robot nel software di simulazione. L'obiettivo è quello di ottenere una risoluzione di visualizzazione dei link "accettabile" che sia la minore possibile. Il criterio decisionale sarebbe la velocità di aggiornamento di visualizzazione da una configurazione robot alla successiva, cosicché risulti una raffigurazione fluida del movimento sebbene la (necessaria) discretizzazione della traiettoria.

L'istogramma "View" a barre orizzontali di colore nero, consente una rapida osservazione dei valori relativi al numero di triangoli, ovvero del numero di facce dei triangoli (faces), assunti dai link. Nell'asse orizzontale infatti è rappresentato il numero di triangoli. Non è importante infatti il valore assoluto assunto dai link, bensì i valori relativi in modo da rendere il più possibile omogenea la risoluzione, tenuto conto anche delle diverse dimensioni dei link. Oltre a questo, è mostrato inoltre la somma totale del numero di facce dei link.

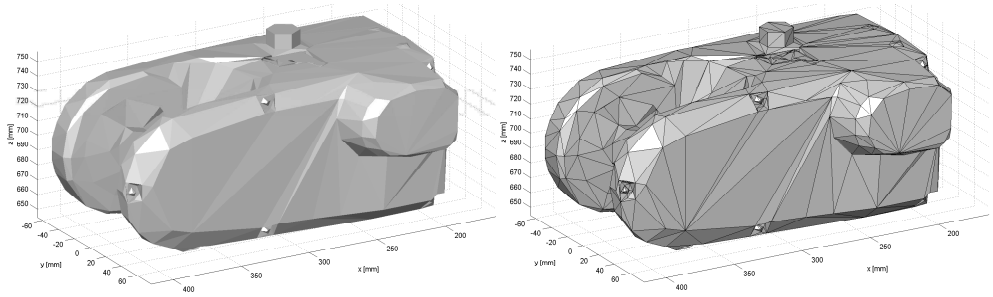
Va precisato infine che la base robot essendo la componente statica del robot, viene rappresentata inizialmente una sola volta. La sua risoluzione di visualizzazione assume quindi un carattere marginale sulla capacità di MATLAB nell'elaborare la simulazione del movimento del robot.

2.3.2 Creazione dei punti per collision detection

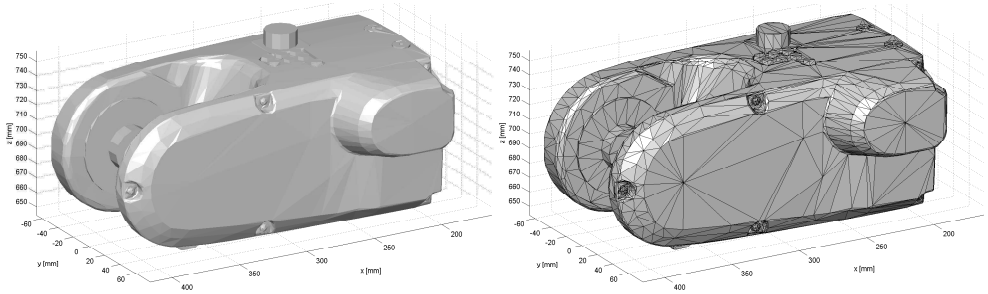
La seconda metà della tabella intitolata "Collision" (riferimento ai campi (B), (E) e (G) di figura 2.21) consente di impostare diversi parametri per la definizione della geometria dei link mediante coordinate di punti. Si possono distinguere due categorie di punti:

- Vertici dei triangoli (rappresentati in blu)
- Punti disposti sulle facce dei triangoli (rappresentati in rosso)

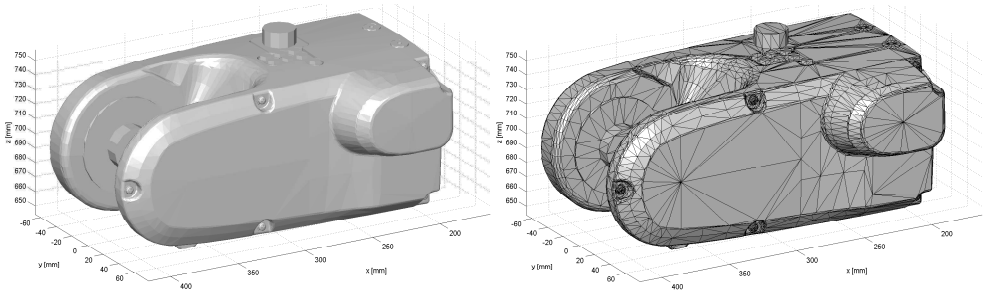
Si è osservato che utilizzando i soli vertici dei triangoli come punti, essi non siano sufficienti alla rappresentazione della geometria dei link a causa della presenza di zone eccessivamente vuote in corrispondenza delle facce dei triangoli. Il problema



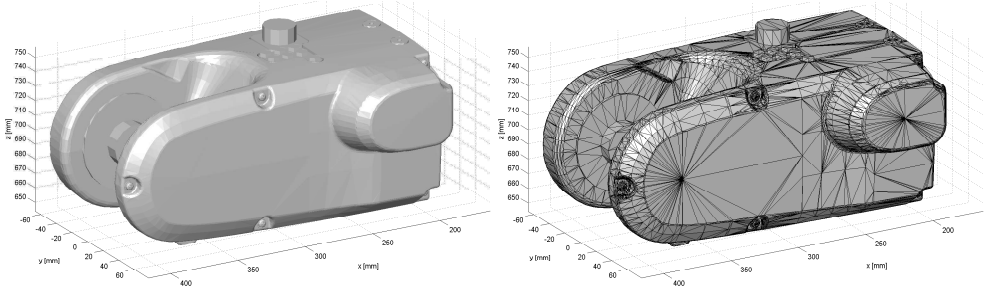
(a) Risoluzione del 5 % (1046 triangoli)



(b) Risoluzione del 20 % (4184 triangoli)



(c) Risoluzione del 40 % (8368 triangoli)



(d) Risoluzione del 100 % (20924 triangoli)

Figura 2.23: Varie risoluzioni di visualizzazione del link 4.

si riscontra generalmente nelle zone in cui la geometria dei link è piana, in quanto i vertici dei triangoli si concentrano ove la geometria presenta zone di curvatura o dettagli costruttivi.

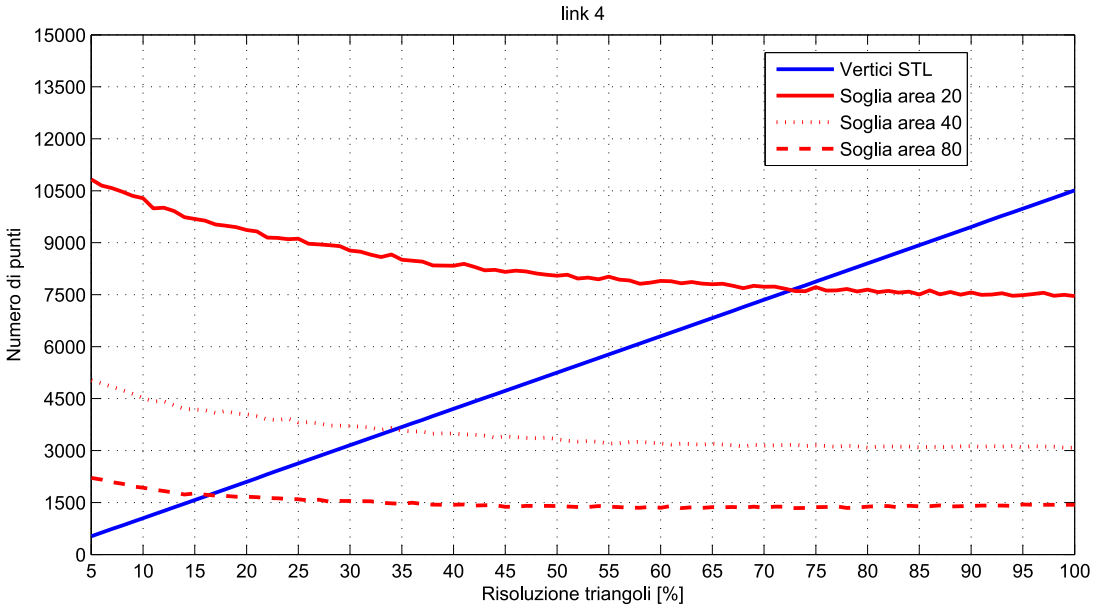


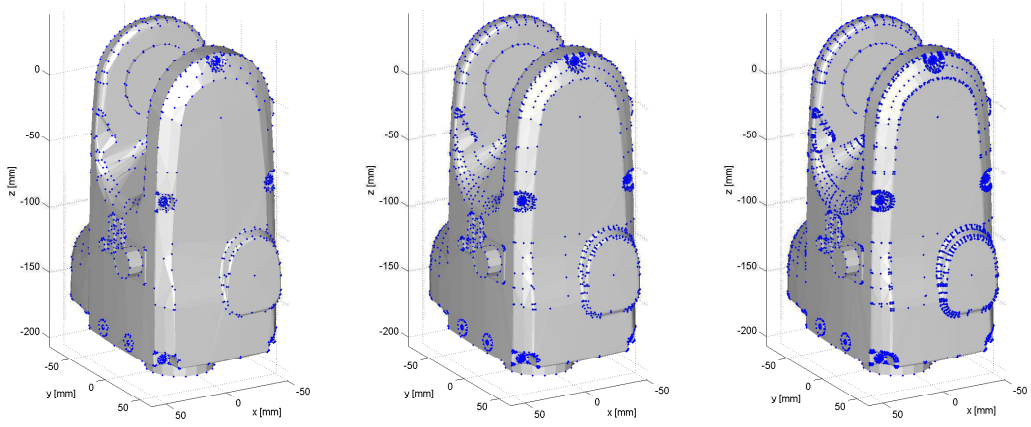
Figura 2.24: Variazione del numero di punti dati da i vertici e tre valori di soglia dell'area in funzione del raffinamento di visualizzazione per il link 4.

Analogamente alla risoluzione della visualizzazione, è possibile variare la percentuale di raffinamento dei vertici mediante uno slider. Aumentando infatti il numero di triangoli, si aumentano di conseguenza anche il numero di punti dati dai vertici. L'aumento avviene ancora una volta in modo proporzionale come confermato dal grafico di figura 2.24.

Va evidenziato che i vertici vengono considerati unici anche se uno stesso vertice appartiene a più triangoli, diversamente dalla descrizione in formato STL che contiene una ridondanza in questo senso.

Un esempio di visualizzazione per diversi valori di raffinamento del link 4 è mostrato in figura 2.25. Si può osservare come aumentando la risoluzione, rimangono comunque delle zone sgombre da punti nella geometria.

Tramite questo parametro si vuole quindi raggiungere un compromesso fra il minor numero di vertici e la miglior descrizione delle zone in cui la geometria è



(a) Impostazione 20% (2096 punti)

(b) Impostazione 40% (4203 punti)

(c) Impostazione 100% (10510 punti)

Figura 2.25: Esempio di raffinamento dei vertici triangoli (punti blu) per diversi valori del parametro.

complessa, in modo da non costituire un'eccessiva ridondanza di punti al fine del collision detection.

2.3.3 Algoritmo di raffinamento dei punti

É stato ideato ed implementato un algoritmo che cerca di occupare in modo omogeneo le zone sgombre di punti nella geometria.

Il principio è quello di suddividere (split) iterativamente i triangoli della rappresentazione in altri 3 o 4 triangoli e considerare i loro vertici come nuovi punti aggiuntivi. A tal fine è possibile impostare 3 parametri, ovvero:

1. Soglia minima dell'area dei triangoli $[mm^2]$
2. Lato minimo dei triangoli $[mm]$
3. Lato massimo dei triangoli $[mm]$

La suddivisione riguarda solo i triangoli con un area maggiore della soglia minima (1) o aventi un lato che è maggiore del lato massimo (3). Di questi si distinguono due tipologie di triangoli: proporzionati o allungati. I primi possiedono tutti i lati che sono maggiori del lato minimo (2) per i quali, ad ogni iterazione,

si determinano altri tre punti aggiuntivi; per quelli allungati invece i triangoli vengono splittati in altri tre triangoli e aggiunti quindi due punti appartenenti ai lati lunghi. Lo schema basilare è quello mostrato in figura 2.26. Un esempio concreto di principio di funzionamento dell'algoritmo è mostrato in figura 2.27.

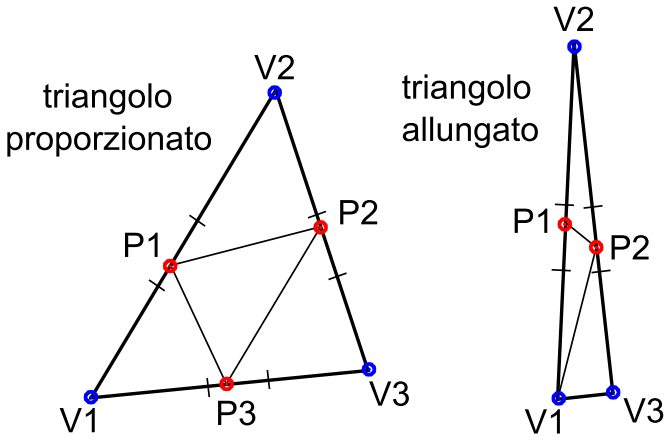
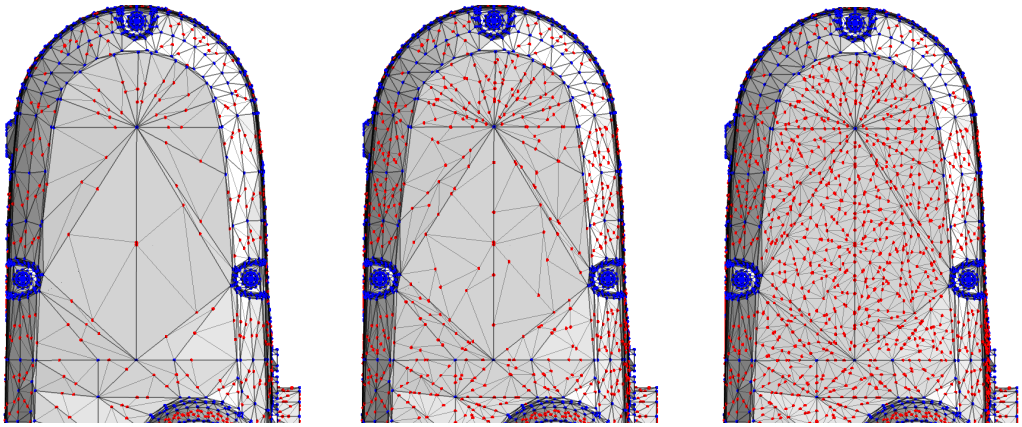


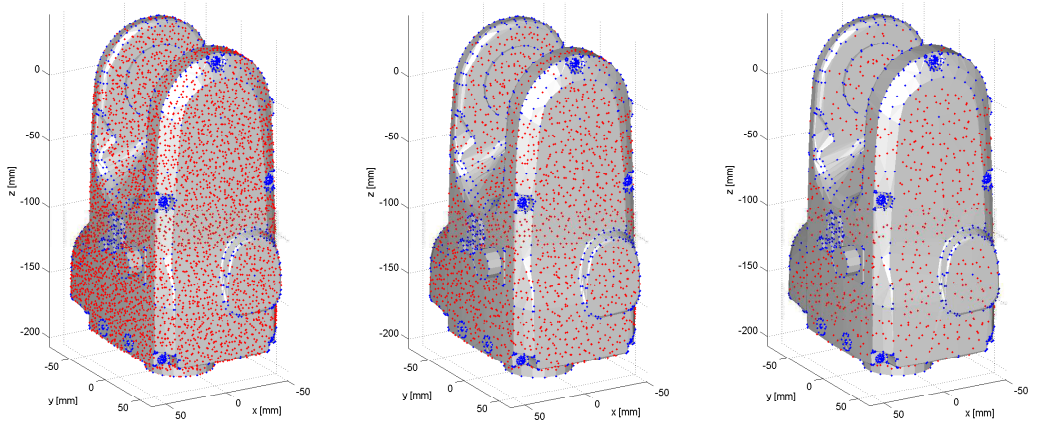
Figura 2.26: Principio base di funzionamento dell'algoritmo.



(a) Prima iterazione (b) Seconda iterazione (c) Risultato finale

Figura 2.27: Funzionamento algoritmo per raffinamento dei punti utilizzando parametri del 40% per il raffinamento vertici (punti blu), soglia area di 20 mm^2, lato corto 45 mm e lato lungo 50 mm (valori di default).

Va osservato che i punti aggiuntivi sono determinati in modo casuale all'interno di un intervallo pari ad un quarto del lato, posizionato nella mediana del lato corrispondente.



(a) Impostazione 20 mm^2 (9444 punti)

(b) Impostazione 40 mm^2 (4026 punti)

(c) Impostazione 80 mm^2 (1686 punti)

Figura 2.28: Effetto causato della variazione della soglia aria minima (punti rossi) utilizzando un raffinamento dei vertici (punti blu) costante del 20%).

La soglia dell'area minima è fra i tre parametri quello influisce maggiormente sul numero di punti che vengono aggiunti. Il grafico di figura 2.24 (pagina 40) mostra l'andamento di questo parametro per tre diversi valori al variare della percentuale di raffinamento dei vertici STL. Si può osservare come il numero di punti tendi a stabilizzarsi su una soglia orizzontale nonostante vengano aumentati i triangoli da splittare; evidentemente non soddisfano le condizioni per essere elaborati dall'algoritmo. L'effetto visivo sul link 4 è quello mostrato in figura 2.28.

L'istogramma "Collision" a barre orizzontali di colore blu e rosso (vedi (E) di figura 2.21), consente una rapida osservazione dei valori relativi al numero di vertici (Vertex) e numero di punti aggiunti (Added Point) assunti dai link del robot. Nell'asse orizzontale è rappresentato il numero di punti. Non è importante solo il valore assoluto assunto dai link ma anche i valori relativi in modo da rendere il più possibile omogeneo il raffinamento dei punti, tenuto conto anche delle diverse dimensioni dei link. Oltre a questo, è visualizzato il numero totale di vertici dei triangoli, il numero totale di punti aggiunti e la loro somma che costituisce l'effettivo numero di coordinate di punti utilizzati per il test di collisione.

2.4 Creazione dei bounding box

Dal menù della GUI, selezionando "Create Link Box", è possibile definire i bounding box per ogni link del robot. L'obiettivo è quello di definire e personalizzare un involucro che racchiuda ogni link costituito dall'intersezione di piani. L'interfaccia è stata realizzata in modo tale da definire i bounding box in modo semi-automatico e consentire anche ad un'utente non esperto di personalizzare e modificare a piacimento la definizione di questi involucri.

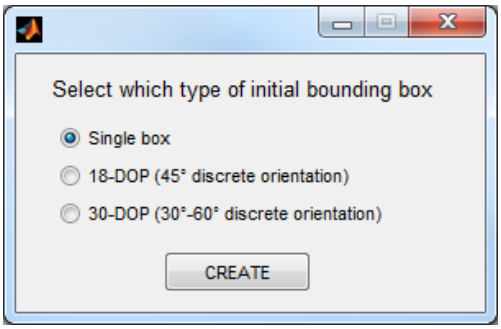


Figura 2.29: Selezione impostazioni iniziali di default.

Alla prima definizione, compare la finestra di figura 2.29 dalla quale è possibile scegliere tre diverse impostazioni di iniziali di default:

1. Singolo box
2. 18-DOP (3 box con assi sfasati di 45°)
3. 30-DOP (6 box con assi sfasati di 30° e 60°)

Si ricorda, come già affermato nel capitolo ..., che il 18-DOP corrisponde ad un box con i 12 lati tagliati a 45° (intersezione di 18 piani).
Mediante 30-DOP (intersezione di 30 piani) invece ognuno dei 12 lati viene tagliato due volte a 30° e 60° causando in genere un miglior tight-fitting del link.
Il singolo box consiste in un OBB dato dall'intersezione di 6 piani.

L'interfaccia che compare a seguito della selezione precedente, per esempio del 18-DOP, è mostrata in figura 2.30. Le parti di cui è composto sono:

- (A) Selezione del link

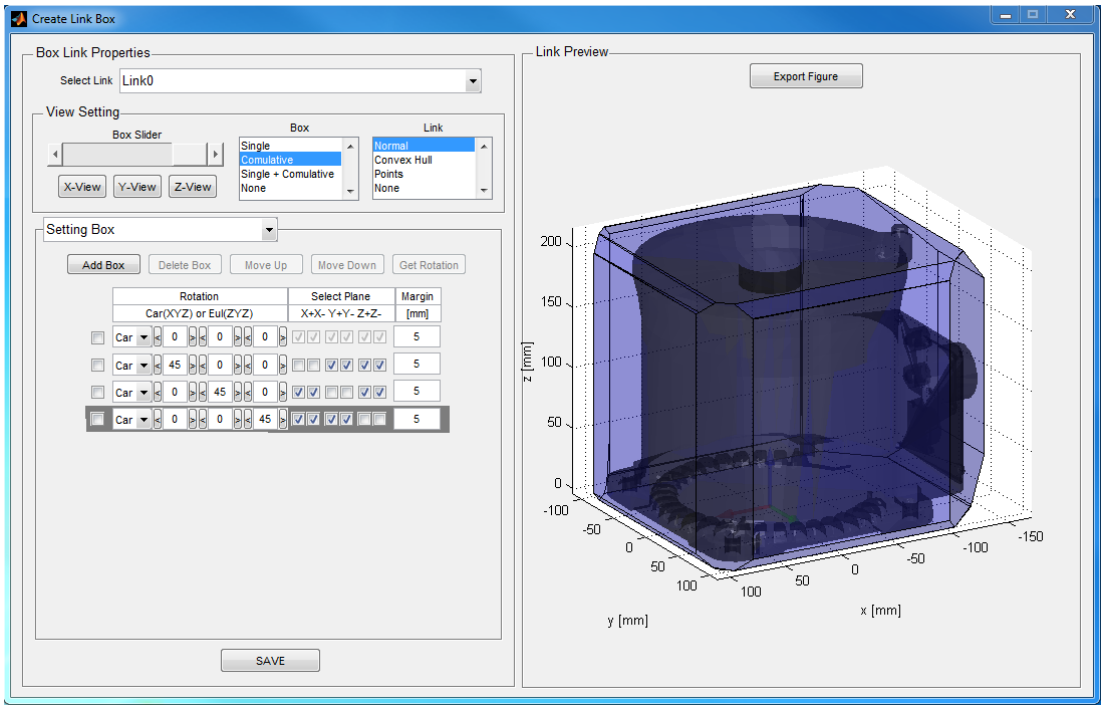


Figura 2.30: Interfaccia principale per creazione dei bounding box.

- (B) Settaggio per l'anteprima di visualizzazione
- (C) Impostare i parametri del box, suddivisi in:
  - "Setting Box" per modificare e personalizzare la definizione dei box
  - "Delete Point" per escludere dei punti dalla definizione dei box
- (D) Anteprima di visualizzazione
- (E) Salvataggio e chiusura dell'interfaccia

2.4.1 Settaggio per l'anteprima di visualizzazione

Tramite il pannello di figura 2.31 è possibile gestire l'anteprima di visualizzazione inerente il link selezionato. Esso è composta da:

- (A) Slider per la selezione del box da visualizzare
- (B) Tipo di visualizzazione del box selezionato
- (C) Tipo di visualizzazione del link selezionato

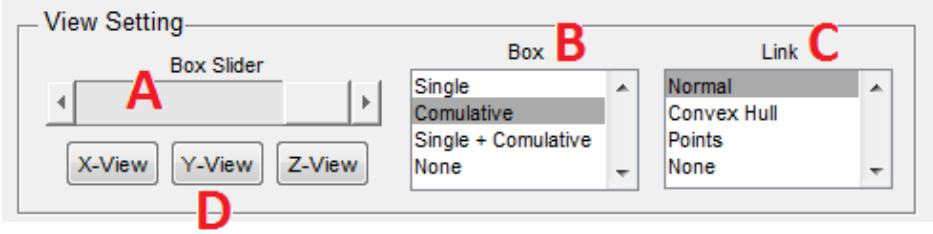


Figura 2.31: Opzioni per settaggio dell'anteprima di visualizzazione.

- (D) Viste prospettiche riferite al box selezionato

Gli step dello slider (A) saranno tanti quanti il numero di box definiti per il singolo link, ovvero pari al numero di righe nella tabella di definizione dei box che verrà discussa nella prossima sezione .... La riga corrispondente al box visualizzato viene evidenziata da un grigio più scuro.

La visualizzazione dei box (B) avviene in modo indipendente da quella del link (C) impostando per entrambi una delle 4 possibili scelte nelle corrispettive liste. Per quanto riguarda il link le possibilità di visualizzazione sono: normale, convoluzione convessa, punti o nessuna visualizzazione. L'effetto di queste selezioni sul link0 (base) è mostrato in figura 2.32.

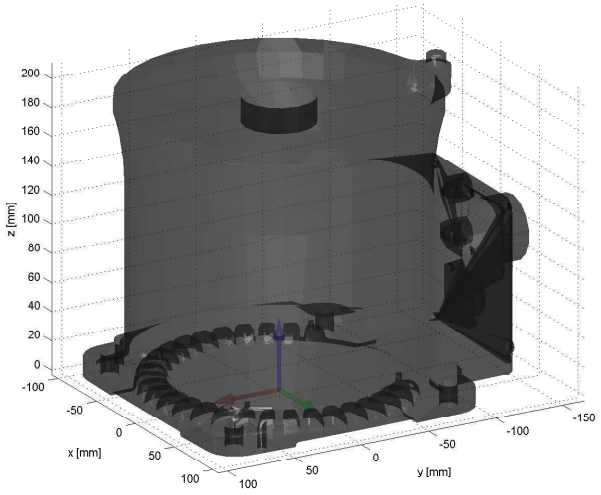
Le possibili visualizzazioni per i box sono: singolo box corrente, intersezione comulativa con i box precedenti, box corrente insieme al box comulativo oppure nessuna visualizzazione del box. Per comprenderne il significato si rimanda alla sezione....

Le viste prospettiche corrispondono ai tre piani YZ ("X-View"), XZ ("Y-View") e XY ("Z-View") riferiti all'orientazione del box corrente selezionato dallo slider. Esse servono ad agevolare l'utente nella visualizzazione e definizione del box.

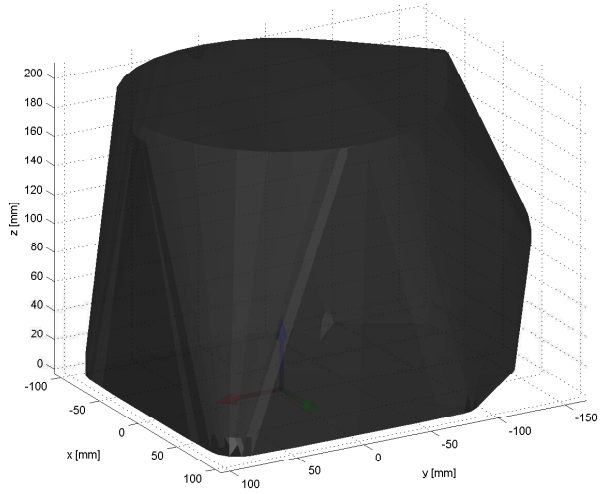
2.4.2 Definizione dei box

Tramite il pannello "Setting Box" di figura 2.33 è possibile avere una panoramica dei parametri e personalizzare la definizione dei box.

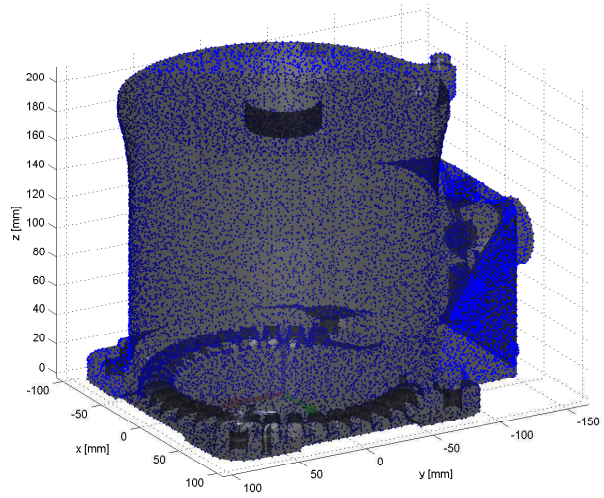
Ogni riga della tabella identifica un box, ovvero un insieme di al più sei piani, a partire dal livello 1 (più basso) dato dalla primo box fino al livello n-esimo



(a) Normal



(b) Convex hull



(c) Point

Figura 2.32: Opzioni di visualizzazione del link (base robot).

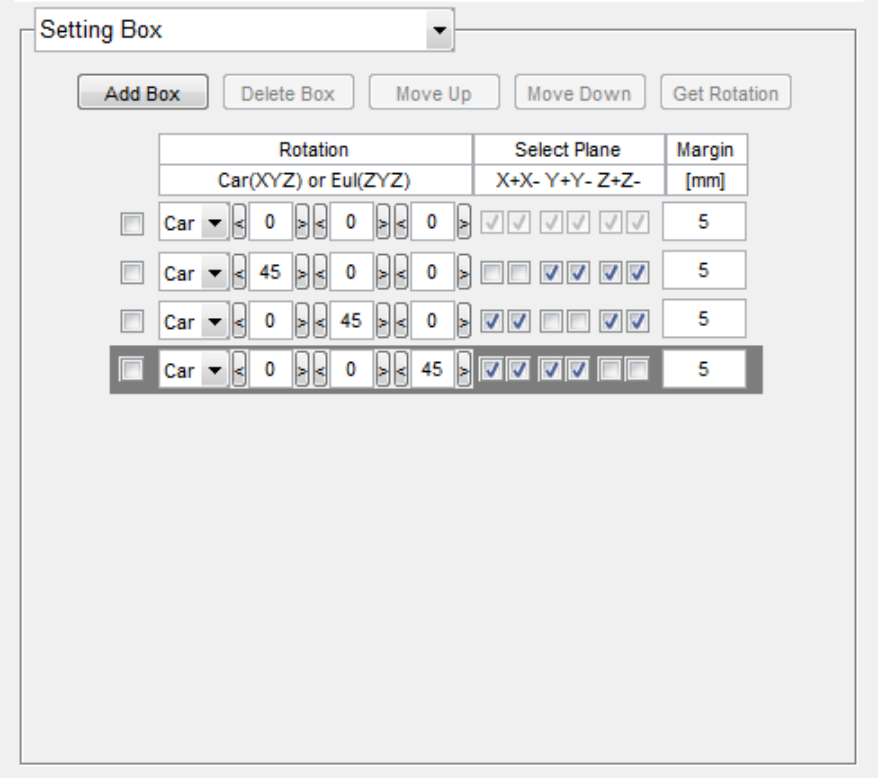


Figura 2.33: Definizione bounding box di default per 18-DOP.

dato dall'ultima riga della tabella. Si parlerà di box comulato di livello i-esimo intendendo l'intersezione dei piani definiti dai box da livello 1 fino livello i. Il bounding box comulato di ultimo livello (dato dall'intersezione di tutti i piani) risulta un poliedro convesso. Esso costituisce il modello geometrico del link corrispondente e potrà essere visualizzato anche dalla GUI principale mediante il pulsante "Boxes".

Ogni box corrispondente ad una riga della tabella è definito da:

- Rotazione di cardano (XYZ) o rotazione di Eulero (ZYZ) rispetto la terna di Denavith-Hartenberg del link
- Selezione dei piani
- Margine di sicurezza

Le dimensioni dei lati del box non sono direttamente modificabili, in quanto vengono implicitamente definite dalla rotazione impartita e calcolate dalla distanza massima e minima dei punti che definiscono il link lungo i tre assi X,Y e Z. É

possibile eventualmente eliminare dei punti come viene illustrato nella sezione..., e le dimensioni del box si aggiorneranno di conseguenza.

Ogni piano appartenente ad una box aggiunge un livello di dettaglio maggiore riducendo il volume del box comulato. Può accadere che alcuni piani siano già stati definiti da box di livello più basso, o che il piano non introduca una significativa riduzione del volume comulato. Ecco il motivo per il quale è possibile scartare dei piani dalla definizione del box. É sufficiente deselezionare i piani che sono stati nominati secondo l'asse normale ed il verso positivo o negativo dell'asse, ovvero: X+, X-, Y+, Y-, Z+ e Z-. Aiutandosi con l'anteprima di visualizzazione questa selezione risulta estremamente intuitiva.

Esistono tuttavia delle limitazioni nella definizione dei box:

- dev'essere definito almeno un box per ogni link
- il box di livello 1 (il primo) deve possedere tutti i sei piani
- è possibile creare al massimo 11 box
- per ogni box dev'essere selezionato almeno un piano

L'ordine con il quale sono definiti i box è importante in quanto determina l'ordine con il quale viene processato il test di collisione del link. Ogni riga della tabella, ovvero ogni livello, comporta una rotazione del sistema di riferimento che aggiunge del tempo per l'esecuzione del collision test.

É possibile impostare infine un margine di sicurezza che di default è di $5mm$. Oltre ad evitare le collisioni con l'ambiente, tramite questo parametro il modello geometrico del link risulta ingrandito assicurando quindi un certo margine di sicurezza.

2.4.3 Visualizzazione dei box

Come anticipato precedentemente, esistono 4 possibili visualizzazioni per i box, ovvero: singolo box corrente, intersezione comulativa con i box precedenti, box corrente insieme al box comulativo oppure nessuna visualizzazione del box.

Ad esempio selezionando il quarto livello del 18-DOP della base robot (impostazioni di figura 2.33) le opzioni di visualizzazione del box sono visualizzate in figura 2.34.

Compaiono due sistemi di riferimento:

- terna di Denavith-Hartenberg del link posizionata nell'origine e visualizzata in trasparenza
- terna di orientazione del box, posizionata nell'origine, con le lunghezze degli assi positivi X+, Y+ e Z+ corrispondenti alle dimensioni del box

Il secondo sistema di riferimento non viene visualizzato selezionando il box comulato.

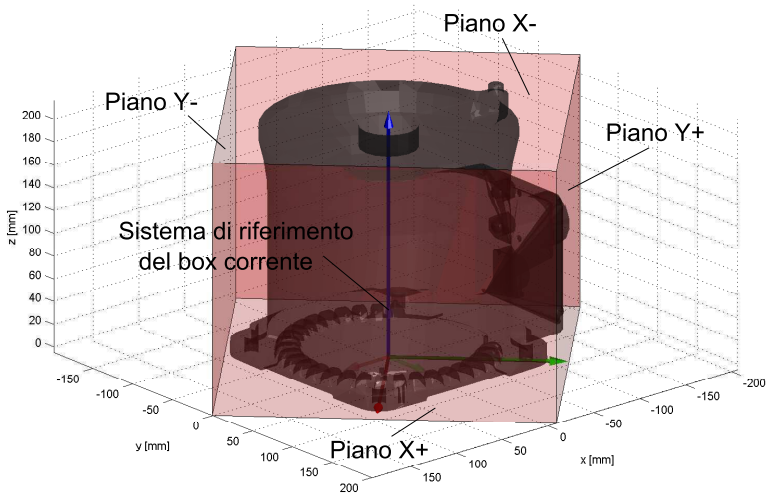
Va evidenziato inoltre che a causa principalmente di errori numerici di approssimazione, l'intersezione dei piani che porta alla visualizzazione del box comulato può contenere dei difetti visivi. L'algoritmo implementato in MATLAB utilizza molti concetti di geometria computazionale e diversamente dalle apparenze risulta molto complesso. Si è scelto quindi un compromesso fra la robustezza del calcolo e la visualizzazione che potrebbe contenere delle piccole imperfezioni.

Le viste prospettiche rivestono un ruolo molto importante al fine di agevolare l'utente nell'impostazione delle rotazioni da impartire al box. Un esempio del loro utilizzo è mostrato in figura 2.35.

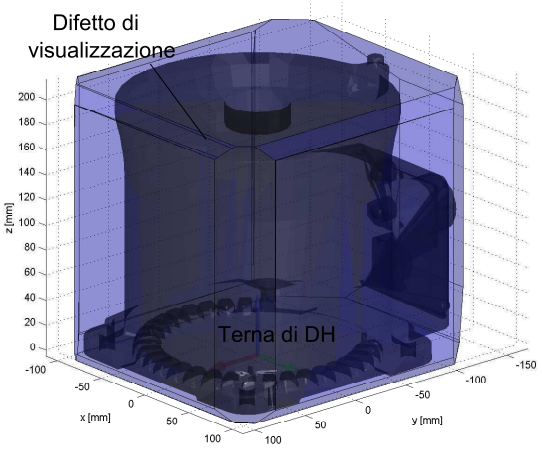
2.4.4 Funzionalità aggiuntive

Le funzionalità aggiuntive permettono una completa personalizzazione della definizione dei box. Selezionando un livello (selezione del checkbox a sinistra delle righe della tabella) si abilitano dei pulsanti (figura 2.36) che consentono di:

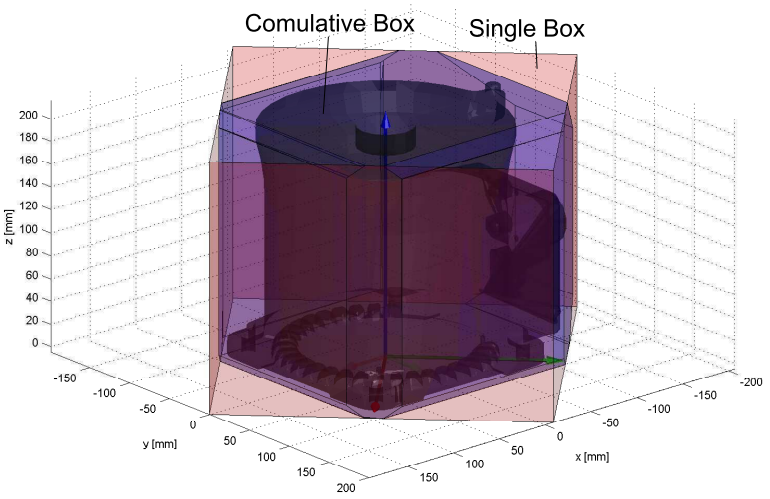
- (A) Aggiungere un box al livello successivo di quello selezionato
- (B) Eliminare il box selezionato
- (C) Shift in alto (livello più basso)
- (D) Shift in basso (livello più alto)



(a) Single box

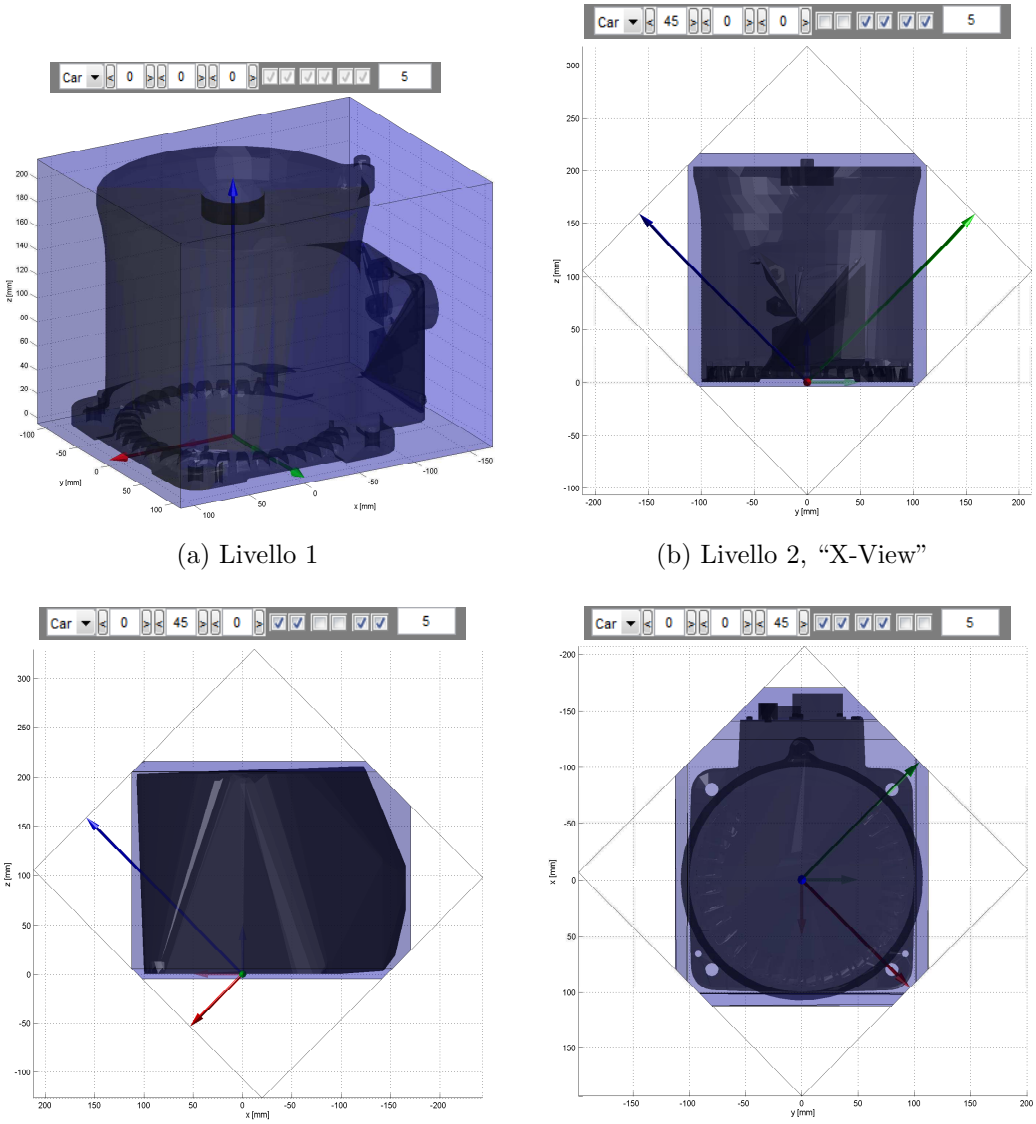


(b) Comulative box



(c) Single + comulative

Figura 2.34: Opzioni di visualizzazione dell'ultimo box (rotazione di Cardano: $R_X(0), R_Y(0), R_Z(45)$, senza i piani Z+ e Z-, margine $5mm$) per 18-DOP riferito al link0 (base robot).



(a) Livello 1

(b) Livello 2, “X-View”

(c) Livello 3, “Y-View”

(d) Livello 4, “Z-View”

Figura 2.35: Viste prospettiche per i vari livelli di un 18-DOP.

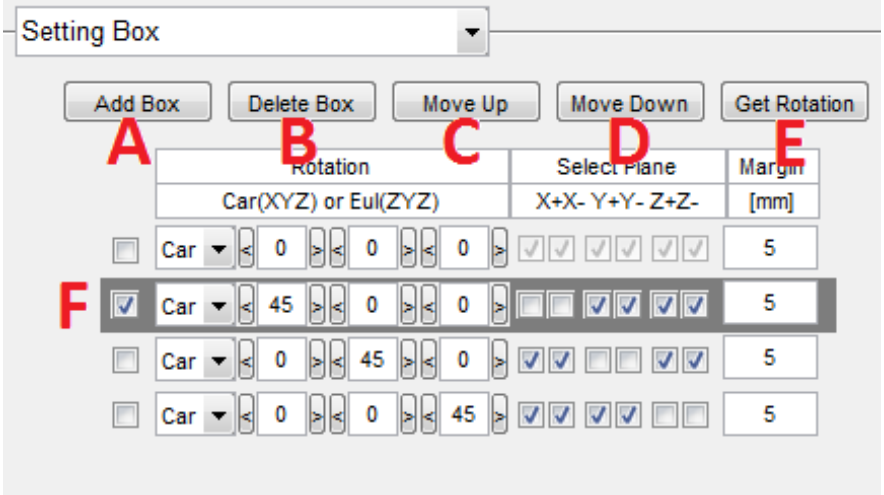
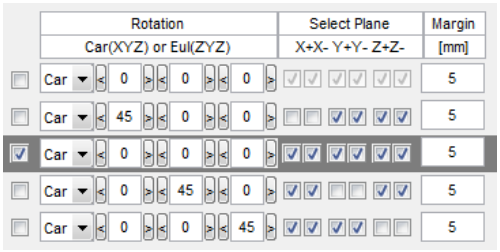
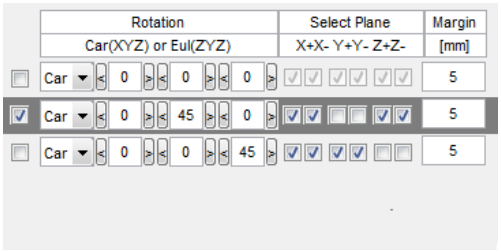


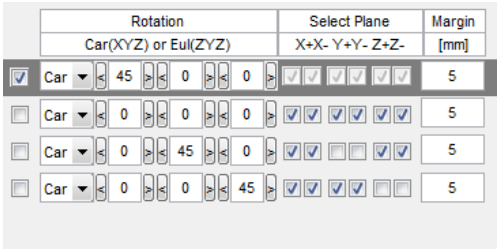
Figura 2.36: Funzionalità di personalizzazione date dai pulsanti.



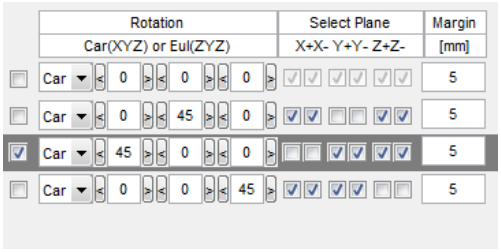
(a) Aggiunta di un box



(b) Eliminazione del box



(c) Shift in alto



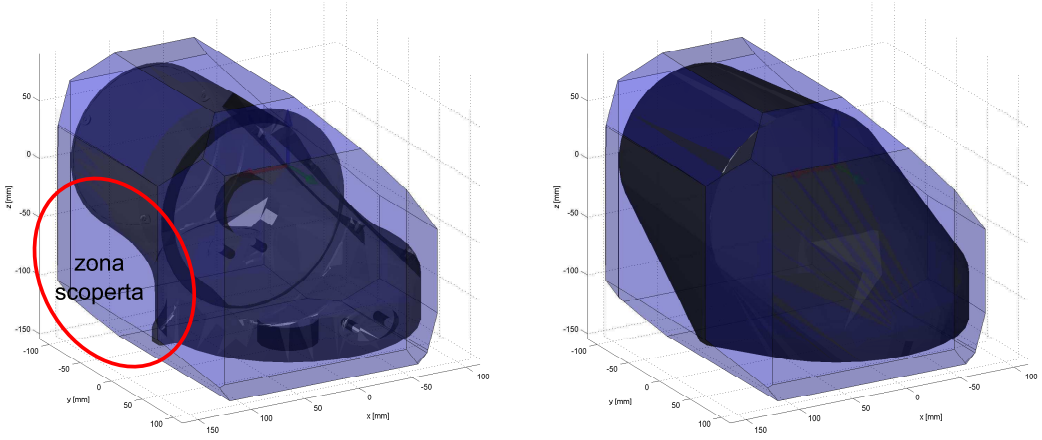
(d) Shift in basso

Figura 2.37: Variazioni causate nella tabella dalla funzionalità aggiuntive.

- (E) Ottenere l'orientamento visualizzato nell'anteprima di visualizzazione

Ipotizzando di selezionare il secondo livello ((F) di figura 2.36), l'effetto delle prime quattro funzioni è mostrato in figura 2.37.

Il pulsante "Get Rotation" può essere utilizzato per definire piani nel caso in cui il bounding box presenti delle zone eccessivamente distanti dalla geometria



(a) Zona scoperta distante dalla geometria (b) Vista come convoluzione convessa

Figura 2.38: 18-DOP del link1.

del link come mostrato in figura 2.38. La convoluzione convessa del link permette di identificare più facilmente queste situazioni.

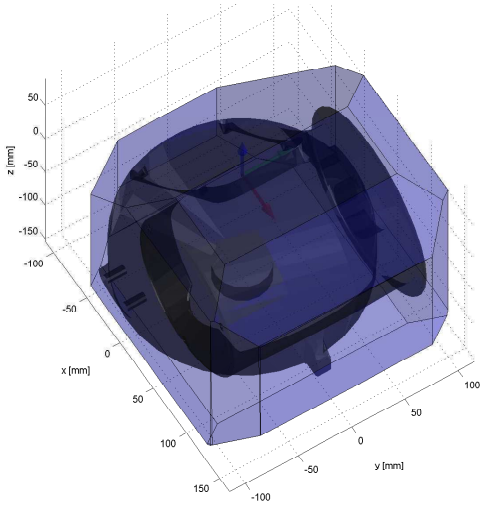
La soluzione a questo problema può essere svolta nel seguente questo modo:

- ruotare l'anteprima di visualizzazione fino alla zona interessata (figura 2.39a)
- aggiungere un box e ottenere l'orientazione tramite l'apposito pulsante (figura 2.39b)
- eliminare eventuali piani che non interessano (figura 2.39c)

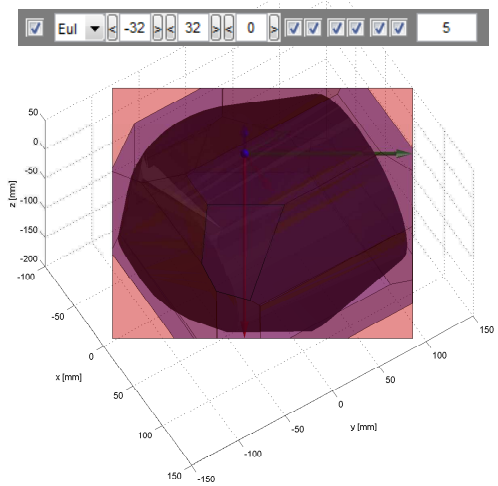
L'orientazione che viene generata fa sempre riferimento ad una rotazione di Eulero, dove l'ultima rotazione rispetto l'asse Z è nulla.

2.4.5 Eliminazione di punti

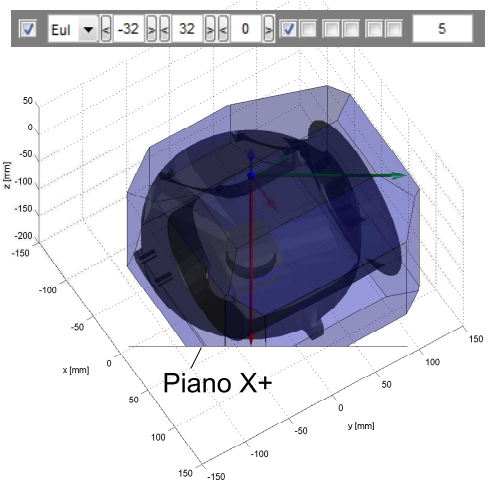
Generalmente succede che delle parti del modello iniziale del link non sono necessarie alla definizione del bounding box. L'interfaccia permette di eliminare queste componenti mediante l'eliminazione di punti. Va ricordato infatti che le dimensioni dei box vengono calcolate dall'orientazione rispetto ai punti e, non potendo quindi definire direttamente le dimensioni del box, eliminare dei punti comporta un miglior tight-fitting della geometria reale del link.



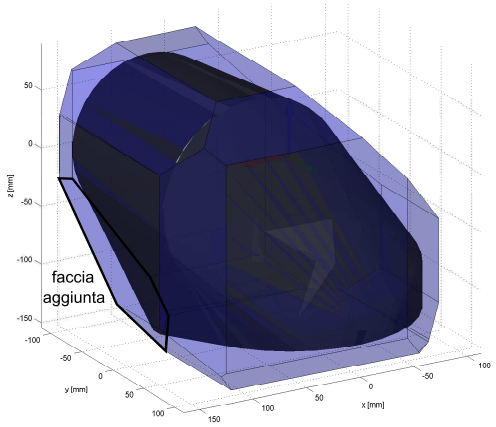
(a) Rotazione della visuale alla zona scoperta



(b) Ottenimento dell'orientazione e del box corrispondente



(c) Eliminazione di eventuali piani



(d) Vista finale della faccia modificata

Figura 2.39: Esempio di applicazione del pulsante "Get Rotation".

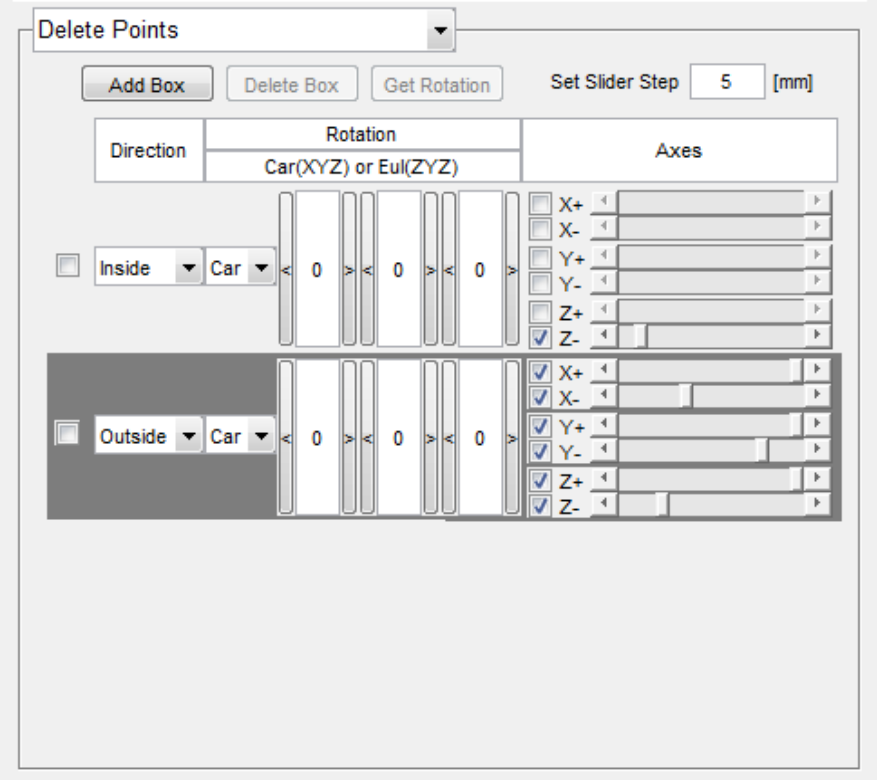
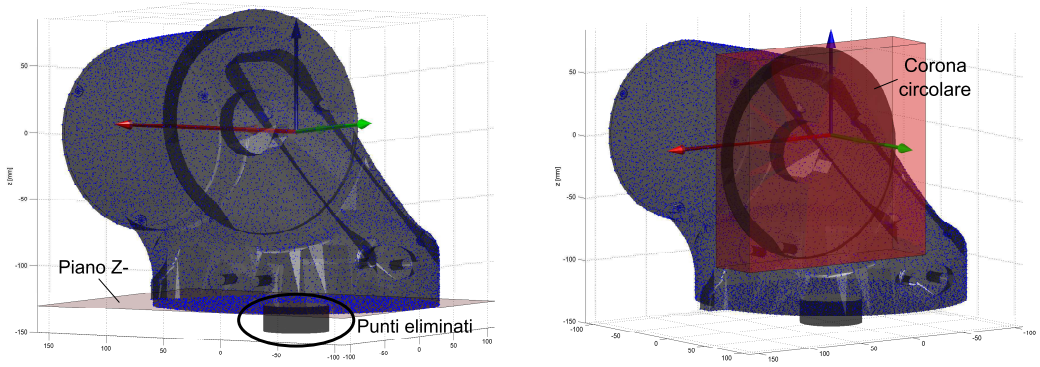


Figura 2.40: Interfaccia per l'eliminazione di punti.

Il pannello "Delete Points" consente questa funzionalità ed è mostrato in figura 2.40. L'esempio riguarda il link1 per il quale sono stati definiti due box il cui effetto è mostrato in figura 2.41.

Il primo box definisce il piano Z- per eliminare i punti del cilindro che essendo interno alla geometria del robot è ovviamente ininfluente nella definizione del bounding box. Il secondo box invece vuole escludere la sottile corona circolare che, pur facendo parte della geometria reale del link, verrà inclusa sicuramente nel modello geometrico del link 3.

Diversamente dalla definizione dei bounding box esposta precedentemente, in questo caso è possibile determinare le dimensioni del box tramite degli slider. Lo step di questi slider vale $5mm$ ed è possibile impostarlo modificando il campo "Set Slider Step". Un'altra differenza è che ogni riga della tabella è indipendente dalle altre, non viene cioè calcolato il box comulato e ognuno determina l'eliminazione di alcuni punti della geometria del link. Non ha senso quindi parlare di livelli e l'unica opzione di visualizzazione consentita per i box è "Single". Infine si può

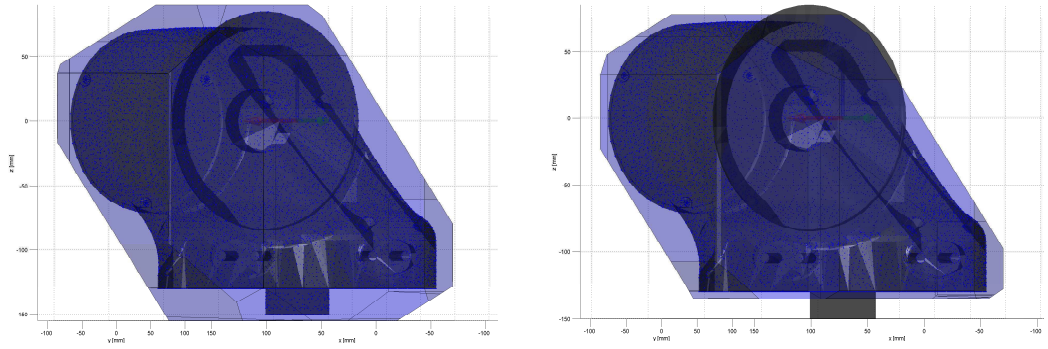


(a) Primo box (b) Secondo box

Figura 2.41: Anteprima di visualizzazione per le impostazioni di figura 2.40.

scegliere di considerare i punti interni od esterni alla definizione del box, mediante l'impostazione della direzione rispettivamente "Inside" oppure "Outside".

Il bounding box del link si aggiorna automaticamente con le nuove disposizioni dei punti, cosi come mostrato in figura 2.42.



(a) Situazione iniziale (b) Dopo l'eliminazione di punti

Figura 2.42: Anteprima di visualizzazione del bounding box del link1.

Le funzionalità aggiuntive date dai pulsanti in questo caso sono solo tre: aggiunta di un box (massimo tre e minimo uno), eliminazione del box e l'ottenimento dell'orientazione. Il loro funzionamento è analogo al caso precedente.

Capitolo 3

Implementazione e simulazione

3.1 Rappresentazione di un piano 3D

Un piano nello spazio tridimensionale può essere pensato come una superficie piana estesa indefinitamente in tutte le direzioni. Esso può essere descritto in diversi modi:

1. Tre punti non collineari (formare un triangolo nel piano)
2. Vettore normale e un punto nel piano
3. Vettore normale e distanza dall'origine

Nel primo caso, i tre punti A,B e C consentono la rappresentazione parametrica del piano P, data da:

$$P(u, v) = A + u(B - A) + v(C - A) \tag{3.1}$$

Negli altri casi la normale al piano è un vettore non negativo perpendicolare ad ogni vettore del piano. Per un dato piano, esistono due possibili scelte di normale, una opposta all'altra. Nel caso di un piano specificato da un triangolo ABC la convenzione è di definire la normale al piano come prodotto scalare:

$$\mathbf{n} = (B - A) \times (C - A) \tag{3.2}$$

Data la normale $\mathbf{n}$ e un punto P appartenente al piano, tutti i punti del piano X possono essere caratterizzati dal vettore $X - P$ perpendicolare ad $\mathbf{n}$, ovvero il

prodotto scalare dei due vettori dev'essere nullo. Questa perpendicolarità risale nella forma implicita del piano:

$$\mathbf{n} \cdot (X - P) = 0 \tag{3.3}$$

Essendo il prodotto interno un operatore lineare che può essere distribuito rispetto la somma e sottrazione, l'espressione 3.3 può essere riscritta nella forma:

$$\mathbf{n} \cdot X = d \tag{3.4}$$

dove $d = \mathbf{n} \cdot P$ rappresenta la *costante-normale* del piano. Quando $\mathbf{n}$ è normalizzato (norma unitaria), d equivale alla minima distanza con segno del piano dall'origine. Se $\mathbf{n}$ non è normalizzato, d comunque rimane la distanza ma nell'unità data dal modulo di $\mathbf{n}$.

Se l'equazione 3.5 è scritta componente per componente, si ricava l'equazione del piano in forma esplicita:

$$ax + by + cz = d \tag{3.5}$$

dove $\mathbf{n} = (a, b, c)$ e $X = (x, y, z)$.

Un piano suddivide lo spazio di due regioni: semispazio positivo che giace nella direzione della normale del piano ed il semispazio negativo situato nella parte opposta del piano. Nel caso 2D una retta suddivide lo spazio un due semipiani (figura 3.1).

3.2 Relazione fra piani e insieme di punti

Si vuole ora determinare se un insieme di punti nello spazio tridimensionale risiedono all'interno di un box costituito dall'intersezione di sei piani. Un esempio di raffigurazione grafica è mostrato in figura...

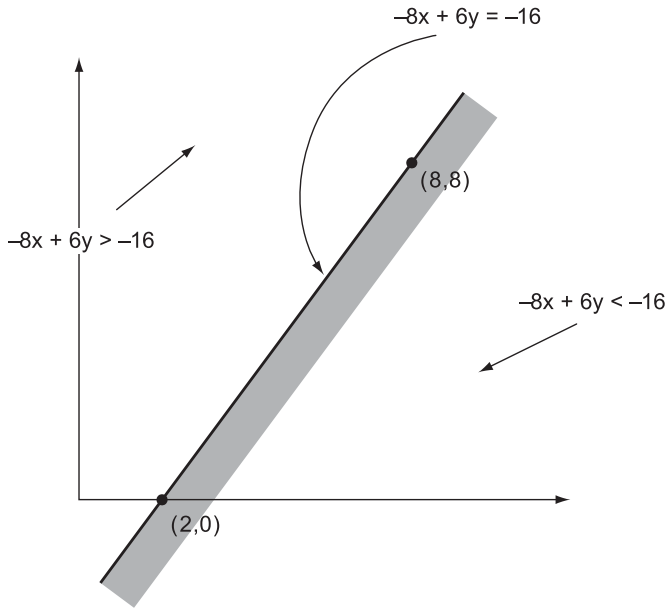


Figura 3.1: La retta $-8x + 6y = -16$ suddivide lo spazio in due semipiani.

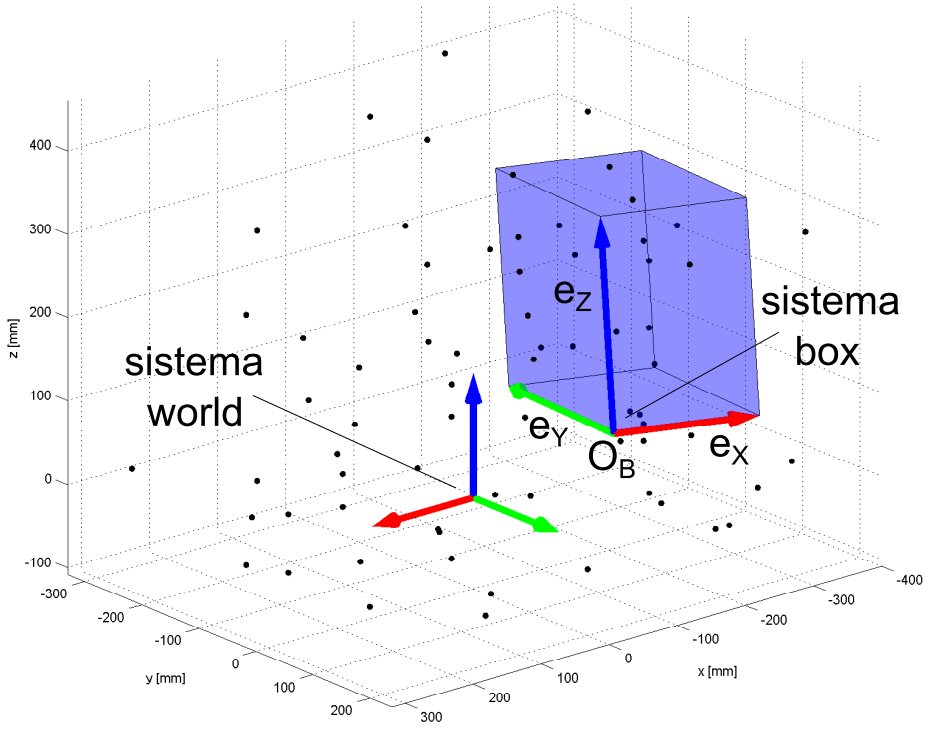


Figura 3.2: Insieme di punti nel sistema di riferimento world e rappresentazione di un box costituito da 6 piani.

Un insieme di p punti rispetto al sistema di riferimento globale (world) è rappresentato da una matrice $4 \times p$ in forma omogenea come:

$$\mathbf{P}_W = \begin{bmatrix} x_1 & x_2 & \dots & x_p \\ y_1 & y_2 & \dots & y_p \\ z_1 & z_2 & \dots & z_p \\ 1 & 1 & \dots & 1 \end{bmatrix} \tag{3.6}$$

dove ogni colonna contiene le componenti del punto i-esimo. La forma omogenea fa riferimento all'ultima riga di "uni" che rende 3.6 compatibile al prodotto matriciale con una matrice di trasformazione.

Un box può essere definito in diversi modi equivalenti come visto nel capitolo 1.1.2. In questo contesto il box è definito rispetto alla terna world mediante:

- Matrice di trasformazione T_{BW} (dimensione 4×4)
- 3 lunghezze dei lati lungo gli assi X_B,Y_B e Z_B

La matrice di trasformazione include la traslazione nell'origine del box O_B e la matrice di rotazione R_{BW} della terna box rispetto terna world:

$$T_{BW} = \begin{bmatrix} & R_{BW} & & (O_B)_W \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

Le equazioni dei piani del box definiti rispetto al sistema world possono essere ricavati facilmente utilizzando come normale dei piani gli assi della terna box (righe della matrice di rotazione R_{BW} normalizzate e aggiustate di segno). Considerando ad esempio la forma esplicita 3.5, si costruisce un sistema di 6 disequazioni del tipo:

$$T_{BW} = \begin{bmatrix} & R_{BW} & & (O_B)_W \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

dove tutti sono termini noti: a_i, b_i, c_i, d_i parametri dei piani e x_i, y_i e z_i le coordinate del punto i-esimo. Quindi un punto (x_i, y_i, z_i) risiede all'interno interno al box se soddisfa simultaneamente tutte le disequazioni del sistema 3.8.

Conclusioni

In questo lavoro di tesi è stata realizzata un'interfaccia grafica con software MATLAB che consente all'utente di importare un modello CAD di un qualsiasi robot a 6 o 4 gradi di libertà. L'output riguarda il modello del robot descritto da nuvola di punti e/o modello geometrico definito da bounding box, utilizzati nella simulazione del sistema di collision detection ideato. Una serie di test simulativi ha consentito di valutare la bontà del sistema. L'analisi dei risultati ha evidenziato che esistono pochi margini per migliorare ulteriormente le prestazioni del sistema di CD realizzato. É stato inoltre ottenuta la massima flessibilità in termini di tipologie di robot importabili ed annessa realizzazione della rappresentazione del modello del robot, sia tramite insieme di punti che modello geometrico.

Appendice A

Matrici di rototraslazione

A.1 Rotazioni assolute e relative

Le matrici di rotazione fondamentali rappresentano delle rotazioni del sistema di riferimento attorno agli assi X, Y e Z.

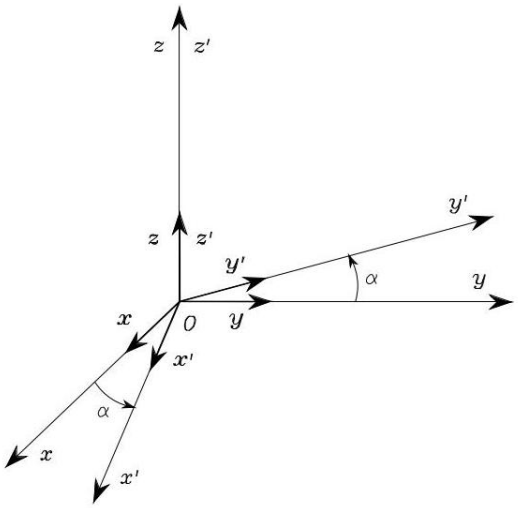


Figura A.1: Rotazione di un angolo α attorno all'asse Z.

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.2}$$

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma & 0 \\ 0 & \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

Le matrici di rotazione fondamentali possono essere composte tra loro per costruire nuove matrici di rotazione che consentano la rappresentazione di relazioni più complesse tra sistemi di coordinate con le origini coincidenti.

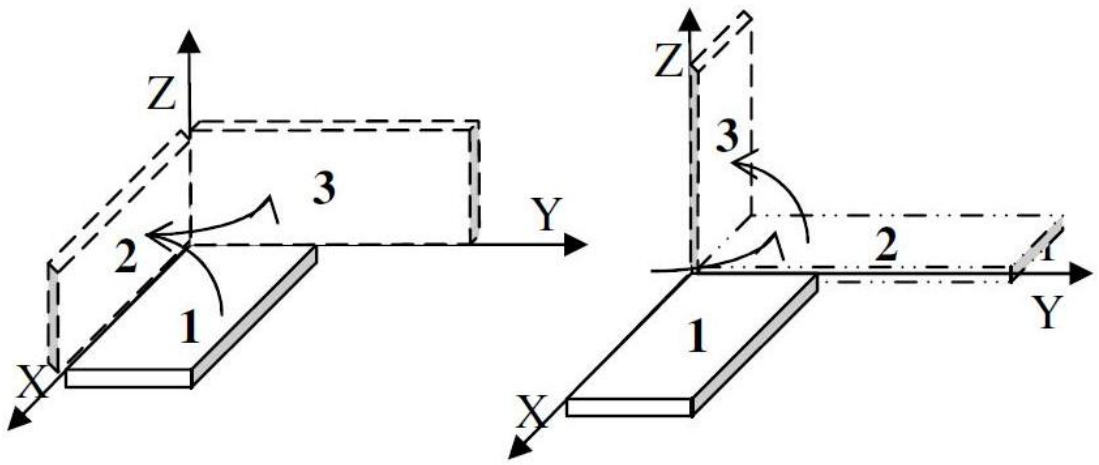


Figura A.2: Rotazioni successive attorno ad assi fissi. Si può notare che non sono commutative.

In particolare, se un sistema di coordinate è ottenuto da una rotazione del sistema di coordinate originale intorno all'asse x di un angolo θ seguita da una rotazione intorno all'asse y di un angolo γ in base alle equazioni relative alle rotazioni fondamentali, le coordinate $\mathbf{q}_1$ e $\mathbf{q}_2$ sono in relazione secondo la seguente matrice di rotazione:

$$\mathbf{q}_0 = R_2^0 \mathbf{q}_2 = R_{x,\theta} R_{y,\gamma} \mathbf{q}_2 = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ \sin\gamma\sin\theta & \cos\theta & -\cos\gamma\sin\theta \\ -\sin\gamma\sin\theta & \sin\theta & \cos\gamma\cos\theta \end{bmatrix} \mathbf{q}_2 \qquad (A.4)$$

La trasformazione inversa da $\mathbf{q}_0$ a $\mathbf{q}_2$ è data dalla matrice $R_0^2 = R_2^{0T}$ trasposta della matrice diretta, e quindi anche sua inversa in quanto matrice ortonormale. Questa proprietà tuttavia non è da confondere con la commutatività di due matrici di rotazione, che non sussiste (vedi figura(C.2)), ad eccezione del caso banale in cui le due rotazioni avvengono intorno allo stesso asse.
Per mostrare che le rotazioni non godono della proprietà commutativa si consideri la rotazione commutata rispetto a quella dell'equazione (C.1), cioè la rotazione di un angolo γ intorno all'asse y seguita da una rotazione di un angolo θ intorno all'asse x. La matrice relativa è data da

$$\mathbf{q}_2 = R_2^0 \mathbf{q}_0 = R_{y,\gamma} R_{x,\theta} \mathbf{q}_0 = \begin{bmatrix} \cos\gamma & \sin\gamma\sin\theta & \sin\gamma\cos\theta \\ 0 & \cos\theta & -\sin\theta \\ -\sin\gamma & \cos\gamma\sin\theta & \cos\gamma\cos\theta \end{bmatrix} \mathbf{q}_0 \qquad (A.5)$$

che evidentemente è molto diversa dalla matrice nell'equazione (C.1).

E' facile dimostrare inoltre, grazie alle proprietà delle matrici ortonormali, che per ottenere una composizione di rotazioni successive attorno ad assi fissi è necessario pre-moltiplicare le varie matrici di rotazione, mentre per ottenere la matrice di rotazione dovuta a successive rotazioni attorno ad assi mobili, ovvero relativi alla rotazione precedente, è necessario post-moltiplicare le varie matrici

di rotazione elementari.
Di conseguenza se applichiamo delle successive rotazioni R_{12}, R_{23} attorno agli assi del sistema di riferimento fisso, al vettore $\mathbf{v}_1$, otterremo $\mathbf{v}_3 = R_{23}R_{12}\mathbf{v}_1$. Sarà dunque $R_{13} = R_{23}R_{12}$ e la sua inversa $R_{31} = R_{21}R_{23}$.
Se invece le rotazioni fossero computate attorno agli assi mobili, l'ordine delle matrici sarebbe stato invertito.

A.2 Angoli di Eulero

La strategia più diretta per la selezione dei parametri minimi descriventi l'orientamento di una terna di riferimento consiste nel caratterizzare la matrice di rotazione in base alla composizione di tre rotazioni successive intorno a tre assi coordinati. I tre angoli associati alle rotazioni vengono denominati angoli di Eulero.
L'arbitrarietà degli angoli di Eulero consiste nel fatto che ciascuna delle tre rotazioni può essere effettuata intorno a un qualsiasi asse coordinato. Tuttavia, condizione necessaria (e sufficiente) perché i tre angoli derivanti da questa caratterizzazione siano indipendenti è che ogni coppia di rotazioni successive avvenga intorno ad assi coordinati diversi.
Vi sono dunque 27 possibili combinazioni di rotazioni, corrispondenti a rotazioni successive intorno ad assi coordinati diversi. Per ogni combinazione, la relativa terna di Eulero viene denominata terna XYZ, o terna YXY, e via di seguito.

La convenzione più usata per gli angoli de Eulero è quella associata alla terna $''ZYZ''$, caratterizzata dalle seguenti operazioni:
1. Rotazione di un angolo α intorno all'asse z;
2. Rotazione di un angolo β intorno all'asse y^I (corrente);
3. Rotazione di un angolo γ intorno all'asse z^{II} (corrente);
La convenzione degli angoli di Eulero prevede che tali rotazioni vengano via via riferite agli assi trasformati secondo l'ultima rotazione effettuata. In base a quanto illustrato nell'equazione (C.1), esse corrispondono dunque a matrici di rotazione che vanno via via a post-moltiplicare le rotazioni precedenti.

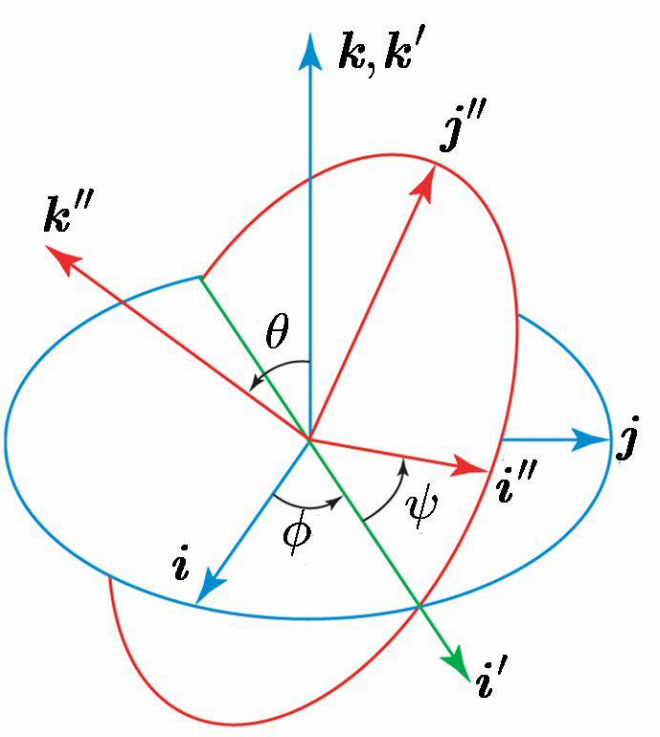


Figura A.3: Angoli di Eulero ZYZ, il sistema fisso XYZ è rappresentato in blu, e il sistema ruotato $X'Y'Z'$ è rappresentato in rosso. La linea dei nodi, indicata con N, è rappresentata in verde.

Considerando dapprima il problema della determinazione della matrice di rotazione R_{ZYZ} a partire dai valori dei tre angoli α, β e γ, si può scrivere la seguente relazione:

$$R_{ZYZ}(\alpha,\beta,\gamma) = R_{Z\alpha}R_{Y,\beta}R_{Z,\gamma} = \begin{bmatrix} C\alpha C\beta C\gamma - S\alpha S\gamma & -C\alpha C\beta S\gamma - S\alpha C\gamma & C\alpha S\beta \\ S\alpha C\beta C\gamma - C\alpha S\gamma & -S\alpha C\beta S\gamma - C\alpha C\gamma & S\alpha S\beta \\ -S\beta C\gamma & S\beta S\gamma & C\beta \end{bmatrix} \tag{A.6}$$

Data una terna α, β e γ di angoli $"ZYZ"$, l'equazione permette di ricavare la matrice di trasformazione corrispondente. Il procedimento inverso (ovvero il calcolo degli angoli corrispondenti ad un determinato orientamento, in base all'espressione della relativa matrice di rotazione R_{ZYZ}) è anche di interesse, ma corrisponde ad un problema algebrico più articolato.

Per la soluzione di quest'ultimo problema è utile introdurre la funzione $(x; y) \rightarrow atan2(x; y)$ che associa ad ogni coppia di ingressi $(x; y)$ un angolo θ tale che $\sin\theta = x/(x^2 + y^2)$ e $\cos\theta = y/(x^2 + y^2)$.
Questa funzione viene anche denominata *arcotangente a 4 quadranti* in quanto, al contrario della classica funzione arcotangente non è soggetta all'indeterminazione tra il primo e il terzo (similarmente, il secondo e il quarto) quadrante. Inoltre, i punti di singolarità in $\pi/2 + k * \pi$ caratterizzanti la funzione classica $atan(x; y)$ non sono presenti in questo caso, grazie al fatto che $atan2(x; y)$ è funzione di due argomenti.
Procediamo ora alla determinazione della trasformazione inversa dell'equazione matriciale espressa sopra. Essendo la matrice di rotazione:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} C\alpha C\beta C\gamma - S\alpha S\gamma & -C\alpha C\beta S\gamma - S\alpha C\gamma & C\alpha S\beta \\ S\alpha C\beta C\gamma - C\alpha S\gamma & -S\alpha C\beta S\gamma - C\alpha C\gamma & S\alpha S\beta \\ -S\beta C\gamma & S\beta S\gamma & C\beta \end{bmatrix} \quad (A.7)$$

Sfruttando le uguaglianze membro a membro e sottraendo una riga all'altra, determiniamo i vari angoli, in particolare

$$\beta = atan2(\sqrt{r_{31}^2 + r_{32}^2}; r_{33}) \quad (A.8)$$

Se $S\beta$ non è $= 0$, allora

$$\gamma = atan2(r_{32}; -r_{31}) \quad (A.9)$$

$$\alpha = atan2(r_{23}; r_{13}) \quad (A.10)$$

Se invece $S\beta = 0$, allora le rotazioni α e γ e avvengono intorno allo stesso asse (eventualmente con verso opposto), quindi si può scegliere arbitrariamente $\alpha = 0$

cosicché $\sin\alpha = 0$ e $\cos\alpha = 1$, per poi determinare

$$\gamma = atan2(r_{21}; r_{11}) \tag{A.11}$$

A.3 Angoli di Cardano

Come gli angoli di Eulero, gli angoli di Cardano (o di Tait-Bryan) rappresentano una successione di rotazioni, ma attorno agli assi coordinati del sistema fisso, non attorno a quelli del sistema solidale.

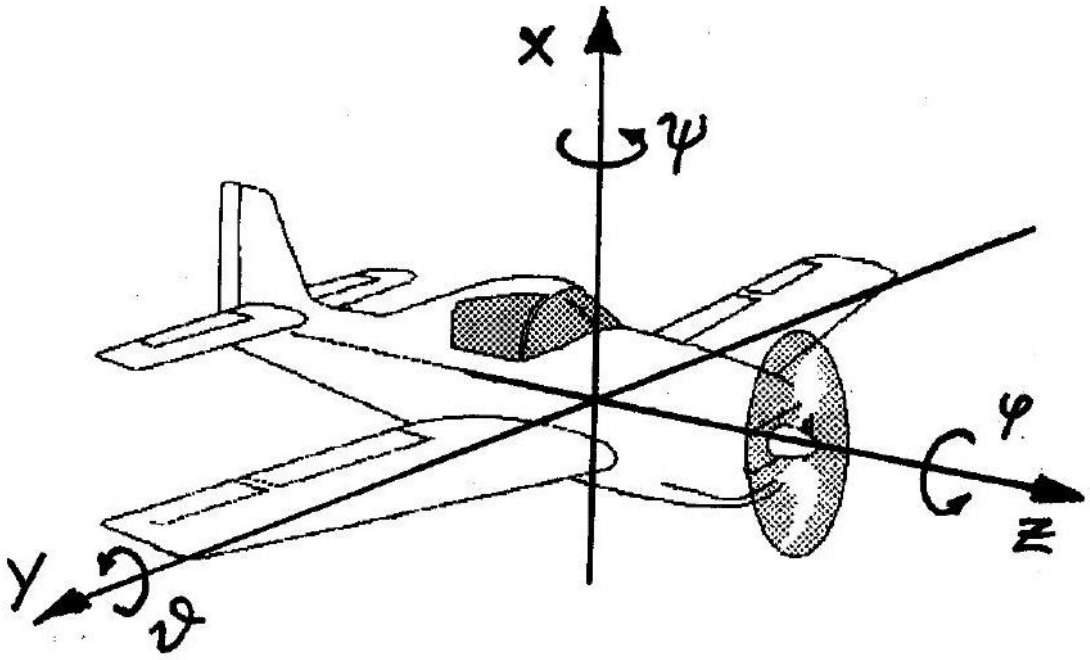


Figura A.4: Rappresentazione degli angoli di RPY

Una convenzione per la rappresentazione minima dell'orientamento particolarmente adottata in campo aeronautico è la convenzione RPY, dove R sta per rollio (*roll*), P sta per beccheggio (*pitch*) e Y sta per imbardata (*yaw*). Questa

convenzione é bene interpretabile facendo riferimento all'assetto di un aeroplano sul quale sia stato fissato un sistema di riferimento il cui asse z è disposto lungo la carlinga, il cui asse y è disposto nella direzione dell'apertura alare e il cui asse x è disposto di conseguenza. Secondo la convenzione RPY, gli angoli di rollio ρ, di beccheggio θ e di imbardata ϕ vengono definiti eseguendo tre rotazioni successive, tutte intorno agli assi del sistema di riferimento originale, secondo la sequenza:

1. Rotazione di un angolo intorno all'asse x;
2. Rotazione di un angolo intorno all'asse y (originale);
3. Rotazione di un angolo intorno all'asse z (originale).

In base a quanto illustrato nel paragrafi precedenti, le tre rotazioni sopra elencate corrispondono a matrici di rotazione che vanno via via a pre-moltiplicare le rotazioni precedenti, infatti eseguendo le rotazioni successive rispetto al sistema di riferimento originale, l'effetto è quello che si otterrebbe se la rotazione in oggetto fosse anteposta a quelle già effettuate.

Pre-moltiplicando, dunque, le matrici di rotazione relative alle trasformazioni sopra elencate, si può procedere alla determinazione della matrice di rotazione $R_{RPY}(\rho;\phi;\theta)$ analoga a quella riportata in equazione A.6 per il caso degli angoli ZYZ:

$$R_{RPY}(\rho,\theta,\phi) = R_{Z\rho}R_{Y,\theta}R_{X,\phi} = \begin{bmatrix} C\rho C\theta & C\rho S\theta S\phi - S\rho C\phi & C\rho C\theta C\phi + S\rho S\phi \\ S\rho C\theta & S\rho S\theta S\phi - S\rho S\phi & C\rho C\theta C\phi + C\rho S\phi \\ -S\theta & C\theta S\phi & C\theta C\phi \end{bmatrix} \tag{A.12}$$

Si osservi che gli angoli di RPY corrispondono ad una delle 27 possibili scelte per gli angoli di Eulero indicate nel paragrafo precedente. In particolare, essi corrispondono agli angoli di Eulero ZYX. La determinazione della trasformazione inversa alla A.12 può essere eseguita in maniera parallela a quanto fatto nel paragrafo precedente per il caso degli angoli ZYZ.

A.4 Rotazione attorno ad un asse

Per questo lavoro di tesi è apparso necessario studiare la composizione di rotazioni che porta un oggetto ed il suo sistema di riferimento, a ruotare attorno ad un asse qualsiasi nello spazio.

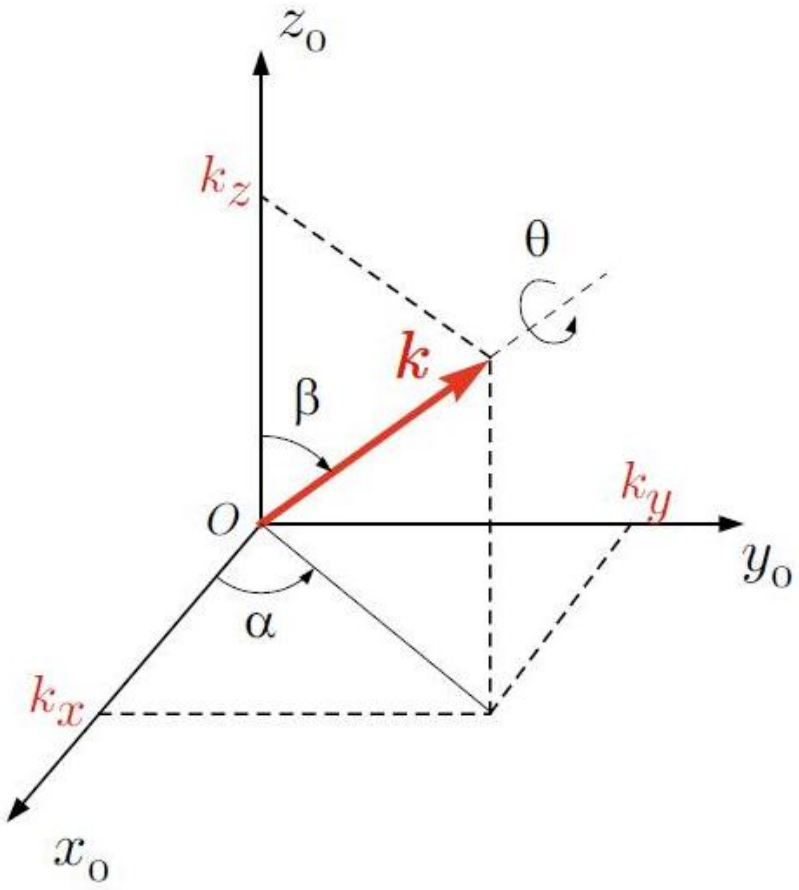


Figura A.5: Rotazione di un angolo θ attorno ad un asse generico.

Una qualsiasi rotazione di un corpo rigido è esprimibile tramite una matrice di rotazione e quindi abbiamo trovato la matrice che esprime la rotazione attorno ad un asse generico.
Con riferimento alla figura, si consideri il caso in cui rispetto alla terna di riferimento $0XYZ$, l'asse di rotazione abbia coordinate $(r_x; r_y; r_z)$, con $(r_x^2 + r_y^2 + r_z^2) = 1$. In altre parole, si supponga che il vettore r mostrato in figura A.5 sia di fatto un versore. In questo caso, la rotazione di un angolo θ intorno all'asse individuato

da r può essere descritta dalla composizione di rotazioni elementari come segue:
1. rotazione di un angolo $-\alpha$ intorno all'asse z per portare l'asse r sul piano verticale individuato dagli assi gli assi x e z;
2. rotazione di un angolo $-\beta$ intorno all'asse y per sovrapporre l'asse r all'asse z;
3. rotazione di un angolo θ intorno all'asse $z = r$;
4. rotazione di un angolo β intorno all'asse y;
5. rotazione di un angolo α intorno all'asse z.

Si noti che le ultime due rotazioni vengono eseguite per riportare l'asse r nella posizione originaria. In particolare, poiché la rotazione da rappresentare avviene intorno all'asse r, tutti i punti dello spazio su questo asse non devono subire nessuno spostamento.

La procedura sopra elencata può essere descritta in termini di matrici di rotazione fondamentali tramite la formula seguente (si noti che tutte le rotazioni sopraelencate sono riferite alla terna fissa dunque corrispondono a matrici di rotazione che vengono via via pre-moltiplicate):

$$R_{r\theta} = R_{Z\alpha} R_{Y\beta} R_{Z\theta} R_{Y-\beta} R_{Z-\alpha} \tag{A.13}$$

nella quale è utile eliminare la dipendenza da α e β, esprimendola in funzione delle componenti di r rispetto al sistema di riferimento $0XYZ$). In particolare, poiché r ha norma unitaria, le seguenti relazioni derivano da semplici argomentazioni geometriche:

$$r_x = \sqrt{r_x^2 + r_y^2} \cos\alpha \tag{A.14}$$

$$r_y = \sqrt{r_x^2 + r_y^2} \sin\alpha \tag{A.15}$$

$$\sqrt{r_x^2 + r_y^2} = \sin\beta \tag{A.16}$$

$$r_z = \cos\beta \tag{A.17}$$

Da queste relazioni si ricava sostituendo seni e coseni di α e β e moltiplicando:

$$R_{r\theta} = \begin{bmatrix} r_x^2(1-C_\theta)+C_\theta & r_x r_y(1-C_\theta)-r_z S_\theta & r_x r_z(1-C_\theta)+r_y S_\theta \\ r_x r_y(1-C_\theta)+r_z S_\theta & r_y^2(1-C_\theta)+C_\theta & r_y r_z(1-C_\theta)-r_x S_\theta \\ r_x r_z(1-C_\theta)-r_y S_\theta & r_y r_z(1-C_\theta)+r_x S_\theta & r_z^2(1-C_\theta)+C_\theta \end{bmatrix} \tag{A.18}$$

che rappresenta la rotazione di un angolo θ della terna originaria intorno all'asse arbitrario r.

Appendice B

Notazione di Denavith-Hartenberg

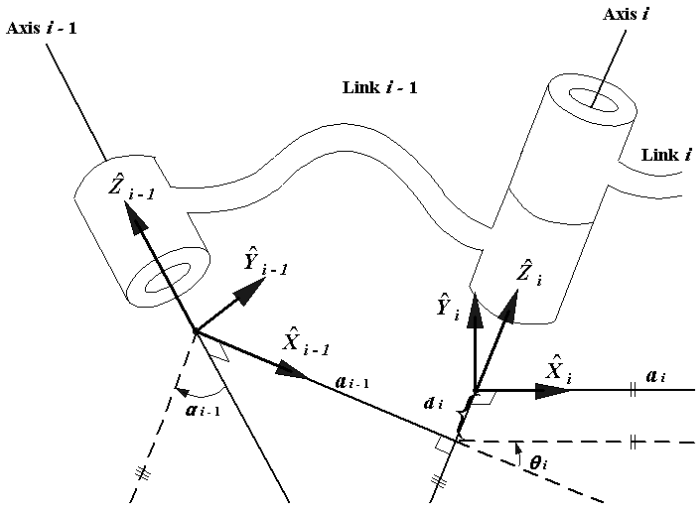
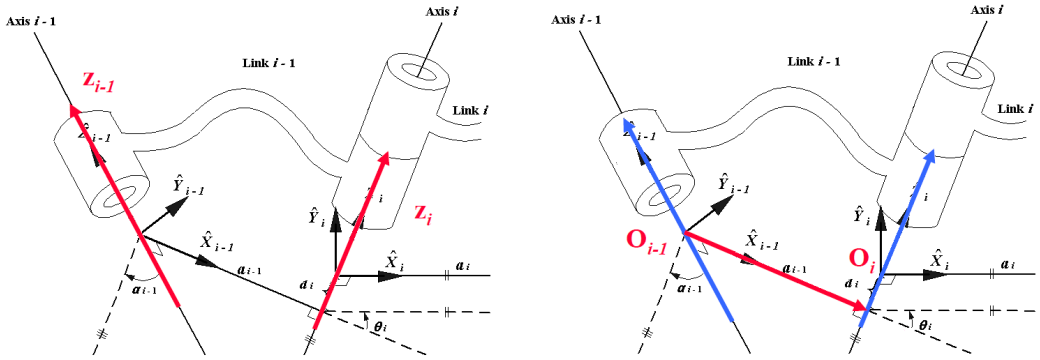


Figura B.1: Relazione fra due due sistemi di coordinate associate ai link.

L'idea è quella di associare ad ogni giunto (link) un sistema di coordinate. Mediante la notazione di Denavith-Hartenberg sono sufficienti 4 parametri per descrivere la matrice di trasformazione dal sistema (i) al sistema $(i-1)$. I parametri riguardano due rotazioni e due traslazioni e sono: α_{i-1}, a_{i-1}, θ_i e d_i.

Per prima cosa è necessario individuare gli assi di rotazione Z_{i-1} e Z_i dei link (vedi figura B.2a). Possono essere orientati arbitrariamente. Si definisce poi X_{i-1}

ortogonale ad Z_{i-1} che a partire dall'origine O_{i-1} intersechi l'asse Z_i in un punto distante d_i dall'origine O_i (vedi figura B.2b). L'asse Y_{i-1} si ricava di conseguenza.



(a) Asse Z_{i-1} e Z_i. (b) Asse X_{i-1} e origine O_{i-1} e O_i .

Figura B.2: Parametri di Denavith-Hartenberg.

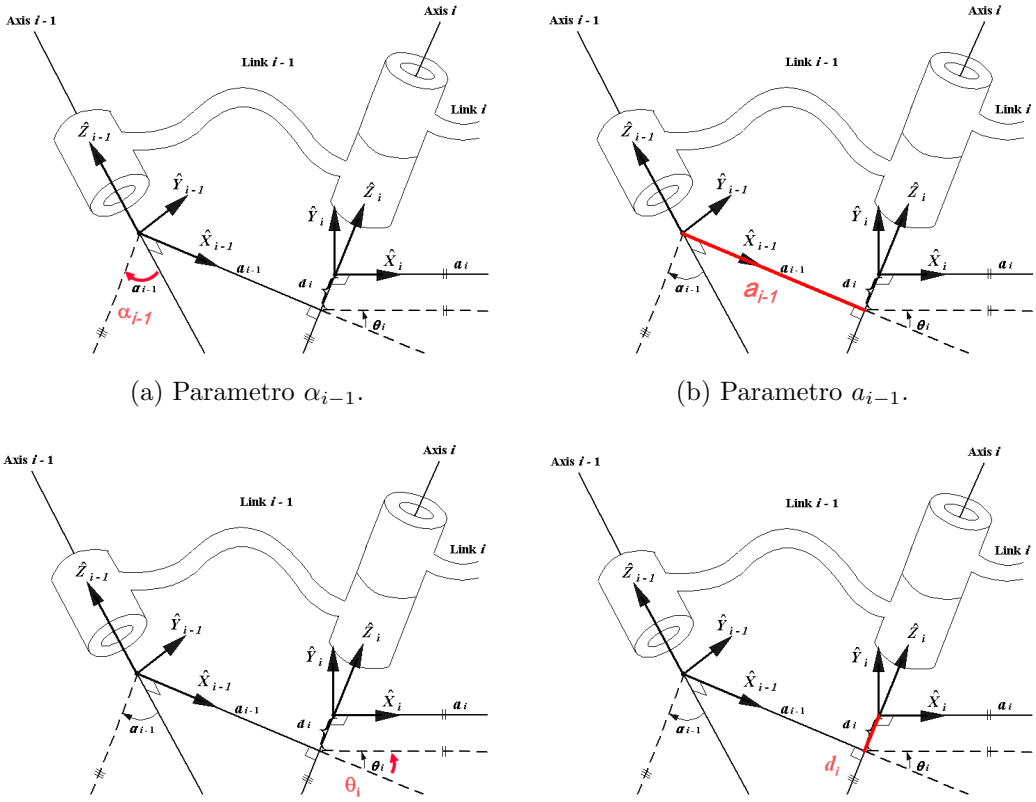
Una volta definite le terne dei link, si passa alla determinazione dei quattro parametri di Denavith-Hartenberg (figura B.3) nel seguente modo:

- α_{i-1} è l'angolo intorno all'asse X_{i-1} che porta l'asse Z_{i-1} lungo la direzione dell'asse Z_i (positivo secondo la regola della mano destra)
- a_{i-1} è la distanza di O_i da O_{i-1} misurata lungo l'asse X_{i-1} (positiva secondo il verso dell'asse X_{i-1})[1]
- θ_i è l'angolo intorno all'asse Z_i che porta l'asse X_{i-1} lungo la direzione dell'asse X_i (positivo secondo la regola della mano destra)
- d_i è la distanza di O_i da O_{i-1} misurata lungo l'asse Z_i (positivo secondo il verso dell'asse Z_i).

La matrice di trasformazione del link i rispetto il link $i-1$ è data da:

$$T_{i,i-1} = T_i^{i-1} = T_{Rx}(\alpha_{i-1})T_{Tx}(a_{i-1})T_{Rz}(\theta_i)T_{Tx}(d_i) = \\ = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & \alpha_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1} * d_i \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1} * d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(B.1)}$$

[1]Se $a_{i-1} = 0$ allora X_{i-1} è normale al piano generato da Z_i e Z_{i-1}



(a) Parametro α_{i-1}.

(b) Parametro a_{i-1}.

(c) Parametro θ_i.

(d) Parametro d_i.

Figura B.3: Parametri di Denavith-Hartenberg.

Bibliografia

[1] M. C. Lin and S. Gottschalk, "Collision detection between geometric models: a survey," 1998.

[2] C. Ericson, *Real-time Collision Detection.* Elsevier, 2005.

[3] T. L. Kay and J. T. Kajika, "Ray tracing complex scenes," 1986.

[4] G. Bradshaw, "Bounding volume hierarchies for level-of-detail collision handing," 2002.