



**Politecnico
di Torino**

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Meccanica LM-33

Tesi di Laurea Magistrale

**"Avris Rotational Scanning": sviluppo di un sistema
automatico per la misurazione ed il collaudo dimensionale**

Relatore

Prof. Massimo Sorli

Studente

Luca Scognamiglio
Matricola: 265988

Tutor aziendali Axist

Fabio Chilò
Ing. Joseph Toma

Luglio 2021

Indice

Indice delle figure	5
Lista acronimi	8
1. Introduzione	9
1.1 Metrologia: stato dell'arte e contesto aziendale.....	9
1.2 Scopo del progetto.....	11
1.3 Analisi strutturale del sistema ARS	12
1.4 Quadro elettrico ed architettura hardware di controllo	16
1.5 Introduzione ai software usati	18
2. Logica di sistema	29
2.1 Configurazione file SA e fase preliminare di calcolo delle posizioni target.	30
2.2 Matlab: calcolo delle posizioni target	35
2.2.1 ScambioDati_SA_ML.m	35
2.2.2 MMR.m.....	35
2.2.3 SMF.m	37
2.3 Protocollo per lo scambio dati tra SA e Matlab	38
2.4 File converter e creazione del file Avris	40
2.5 Avris: albero ed interfaccia utente	41
2.6 Configurazione della cartella "WorkFolder_ARIS".....	42
2.7 Controllo sistema in ambiente TwinCAT-Beckhoff.....	44
2.7.1 Main	45
2.7.2 fbAxis (FB)	46
2.7.3 fbCycleARS (FB).....	47
2.7.4 Gestione delle posizioni target	49
2.7.5 fbAdsIO.....	50
2.8 Interfaccia HMI interna al progetto PLC	51
3. Modello cinematico ARS	55
3.1 Implementazione della cinematica in AVRIS	61
3.1.1 Manipolatore.....	61
3.1.2 End-effector	63
3.1.3 Tracking System.....	64
3.2 Implementazione del modello cinematico nel file master di SA....	65
3.3 Configurazione temporanea del quinto giunto	67

3.4	Individuazione dei parametri cinematici calibrati	68
3.5	Analogie con altri strumenti di misura e bracci robotici	72
4.	Gestione della rotazione del montante	75
4.1	Modello cinematico e di calcolo.....	76
4.2	Integrazione del modello cinematico nello script Matlab	79
4.3	Considerazioni geometriche	83
4.4	Ottimizzazione delle correzioni calcolate per il generico piano di scansione	84
4.5	Scostamenti massimi dal valore nominale causati dal fitting e valutazione del range di merge	87
4.5.1	Primo caso: rotazione montante $\alpha < \alpha_f$ e raggio di manovella $m < m_f$	87
4.5.2	Secondo caso: rotazione montante $\alpha < \alpha_f$ e raggio di manovella $m > m_f$	91
4.6	Gestione della correzione ed effetti nelle target position	92
5.	Conclusioni e sviluppi futuri	95
5.1	Test script SA-ML.....	95
5.2	Test protocollo di comunicazione SA-Matlab	95
5.3	Test modello cinematico implementato su AVRIS.....	95
5.4	Test ciclo macchina nel PLC.....	97
5.5	Risultati sulla scansione a valle della calibrazione	98
5.6	Sviluppi futuri	99
	Bibliografia	103

Indice delle figure

Figura 1.1: (a) CMM Universal della coord3; (b) braccetto di misura Faro, in uso presso Axist; (c) laser tracker Leica AT402 in uso presso Axist.....	9
Figura 1.2: esempi applicativi di Robot Avris	10
Figura 1.3: Struttura Gudel, prodotto TM-40/EP-3 con opzione C	12
Figura 1.4: Sistema montato e cablato nel reparto di ricerca e sviluppo	12
Figura 1.5: sistema di riferimento base adottato e movimentazioni possibili.....	13
Figura 1.6: asse ed azionamento traslazione tavola rotante	14
Figura 1.7: sistema di riferimento per il calcolo della capacità di carico	14
Figura 1.8: Collegamento carrello-montante e alloggiamento riduttore planetario.....	15
Figura 1.9: schema del meccanismo interno del riduttore Teijin RV-E series	15
Figura 1.10: PC-Beckhoff C6930-0060	16
Figura 1.11: terminali Beckhoff.....	17
Figura 1.12: Drive Kollmorgen.....	17
Figura 1.13: Adattatori di segnale dell'encoder rotazione tavola.....	17
Figura 1.14: (a) Blocco alimentazione; (b) Switch	18
Figura 1.15: software usati durante la routine di scansione.....	18
Figura 1.16: scansione TwinCAT PLC.....	19
Figura 1.17: schema macchina a stati	20
Figura 1.18: implementazione della macchina a stati in linguaggio ST tramite l'istruzione CASE.....	21
Figura 1.19: configurazione generica di un azionamento Beckhoff	22
Figura 1.20: Modulo Motion della solution ARS	22
Figura 1.21: task principale del progetto PLC	23
Figura 1.22: (a) sopra l'editor per dichiarare le variabili, sotto istanziamento di FB e richiamo azioni. (b) Stati iniziali del ciclo macchina implementato	23
Figura 1.23: comunicazione ADS tra Server e Client.....	24
Figura 1.24: configurazione generica di un sistema di misurazione con SA.....	25
Figura 1.25: interfaccia di SA.....	26
Figura 1.26: MPs Editor.....	27
Figura 2.1: processo di scansione del sistema Avris Rotational Scanning	29
Figura 2.2: file master aperto dal Workfolder e suite ARS con tasto di avvio script ..	30
Figura 2.3: Interfaccia utente dello script di SA, rename file	30
Figura 2.4: (a) Interfaccia utente dello script di SA per la verifica del posizionamento componente. (b) Interfaccia utente dello script di SA, finestra di recap.....	31
Figura 2.5: esempio di scansione con overlapping	32
Figura 2.6: Piano di intersezione e vettori normali uscenti creati sui vari livelli di scansione al termine dello script ARS	32
Figura 2.7: Albero di SA una volta terminata l'esecuzione dello script ARS.....	33
Figura 2.8: Diagramma di flusso dello "Script_ARs".....	33
Figura 2.9 Arresto esecuzione del codice per assenza del CAD nella cartella di lavoro e warning causa sfornamento range	34
Figura 2.10: Diagramma di flusso function MMR.m	36
Figura 2.11: Applicazione della matrice di rotazione sul generico vettore	36
Figura 2.12: Protocollo di comunicazione tra SA e Matlab.....	38

Figura 2.13: File testo generato da Matlab con le target position da passare agli assi	40
Figura 2.14: Esempio di file in formato Avris	40
Figura 2.15: Interfaccia utente Avris	41
Figura 2.16: Workfolder_ARs	43
Figura 2.17: Interfacciamento PC-Server e PC-Client	44
Figura 2.18: Main della solution	45
Figura 2.19: Dichiarazioni degli assi come oggetti di tipo fbAxis	46
Figura 2.20: macchina a stati, action_AutoRun	48
Figura 2.21: processo di sovrascritture delle variabili di target PLC	49
Figura 2.22: HMI nella configurazioni iniziale	51
Figura 2.23: HMI ad homing effettuato	52
Figura 2.24: HMI movimentazione di fine ciclo	52
Figura 3.1: schema guide Gudel con movimentazioni effettivamente usate	56
Figura 3.2 -Schema del modello cinematico	57
Figura.3.3- Frames secondo la convenzione DH	58
Figura 3.4: A sinistra terna centro trapezio LMI, a destra terna centro trapezio Avris	60
Figura.3.5: Script di verifica del modello cinematico	60
Figura 3.6 - Robotic System, workcell di Avris	61
Figura 3.7 - Kinematics interface, workcell di Avris	62
Figura 3.8 – End-Effector, workcell di Avris	64
Figura 3.9 : (a) configurazione del sistema ARS, (b) configurazione usata per robot Avris	65
Figura 3.10: interfaccia SA per l'inserimento dei parametri cinematici di DH	66
Figura 3.11: Frame nominali dei rispettivi componenti e Teach Pendant interno	66
Figura 3.12: configurazione provvisoria J5	67
Figura 3.13: (a) Leica Absolute Tracker AT402, (b) Red Ring Reflector Leica	69
Figura 3.14: posizionamento dei porta target nei pressi dello strumento di misura, laser e scale bar di taratura	69
Figura 3.15: Field check, scale bar check e barra di calibrazione Axist AXS777	70
Figura 3.16: processo di best fitting per la determinazione dell'asse di rotazione montante	71
Figura . 3.17: Zeiss Pro T Premium, CMM a braccio orizzontale	72
Figura 3.18: topologie di manipolatori usati in ambito industriale	73
Figura 4.1: angoli di incidenza limite durante la scansione, Gocator LMI 2430	75
Figura 4.2: modello cinematico, (a) configurazione nominale e (b) configurazione con rotazione montante	77
Figura 4.3: costruzione geometrica	77
Figura 4.4: logica di calcolo	79
Figura 4.5: data set falso manovellismo con passo di alpha a 0.1	81
Figura 4.6: data set falso manovellismo con passo di alpha a 0.5	81
Figura 4.7: a sinistra valori di alpha con passo pari a 0.1, a destra valori di alpha calcolati in un range continuo	82
Figura 4.8: fenomeno del "distacco" biella-manovella	83
Figura 4.9: Regressione polinomiali con elementi randomici presi dal dataset	84
Figura 4.10: risultati della regressione polinomiale	85
Figura 4.11: schema a blocchi dell'algoritmo di fitting SMF	85

Figura 4.12: risultati ottenuti dalla function SMF	86
Figura 4.13: evoluzioni delle matrici intermedie durante l'algoritmo SMF	86
Figura 4.14: Rappresentazione grafica del caso $\alpha < \alpha_f$ e raggio di manovella $m < m_f$	87
Figura 4.15: particolare della figura precedente	88
Figura 4.16: costruzione grafica A.....	89
Figura 4.17: costruzione grafica per il calcolo di AC"	90
Figura 4.18: caso rotazione montante $\alpha < \alpha_f$ e raggio di manovella $m > m_f$	91
Figura 4.19: zoom figura precedente	91
Figura 4.20: rappresentazione grafica del calcolo della posizione target "traslazione tavola"	93
Figura 4.21: risultati dell'applicazione della SMF.m sul file TargetPosition.txt	93
Figura 5.1: NC-axes, Online view (sx), Function view (dx)	96
Figura 5.2: soffietto utilizzato per la fase di testing.....	96
Figura 5.3: buffer statico delle target position usato in fase di testing	97
Figura 5.4: confronto acquisizione prima e dopo la calibrazione	98
Figura 5.5: caratteristiche geometriche prima e dopo la calibrazione	99
Figura 5.6: HMI del Robot AVRIS.....	100

Lista acronimi

ARS	Avris Rotational Scanning
SA	SpatialAnalyzer
ML	Matlab
TC	TwinCAT
PLC	Programmable logic controller
ST	Structured Test (linguaggio PLC)
FUN	Function
FB	Function Block
PRG	Program
ADS	Automation Device Specification
CMM	Macchine di misura a coordinata
HMI	Human Machine Interface
DH	Parametri di Denavit Hartenberg
GDL	Gradi di libertà
LAN	Local Area Network
MPE	Maximum Permissible Error
CAD	Computer Aided Design
MMR.m	Funzione Matlab
SMF.m	Funzione Matlab

1. Introduzione

1.1 Metrologia: stato dell'arte e contesto aziendale

Al pari di altre fasi tipiche dei processi di fabbricazione, la metrologia riveste un ruolo fondamentale in ambito industriale. Quasi tutti gli oggetti o prodotti richiedono attualmente una misurazione o un collaudo dimensionale, atti a garantire la conformità del pezzo rispetto al disegno realizzato in fase di progettazione. Questo permette al produttore di assicurare la compatibilità tra componenti provenienti da diversi cicli di produzione e al contempo permette al cliente di verificare facilmente le caratteristiche del prodotto tramite il foglio di collaudo. Per questi motivi, e grazie anche alla richiesta di maggiore automazione proveniente dall'industria 4.0, la metrologia è un settore in continuo sviluppo e proiettato al futuro. L'esigenza di ridurre al minimo i tempi di misurazione, senza incidere sulla validità della misurazione stessa, e la volontà di inserire in maniera sempre più integrata il processo di misurazione nelle linee di produzione ha portato e porta tuttora allo sviluppo di nuovi sistemi di misura. Partendo infatti dai primi sistemi della metrologia classica quali calibri, micrometri, comparatori si è passati ad altri sistemi più performanti e spesso accompagnati da software di collaudo ed interfacce dedicate. Tra questi sistemi vi sono sicuramente le CMM, i laser tracker, i sistemi ottici e multisensore ed i braccetti meccanici di misura, figura 1.1.

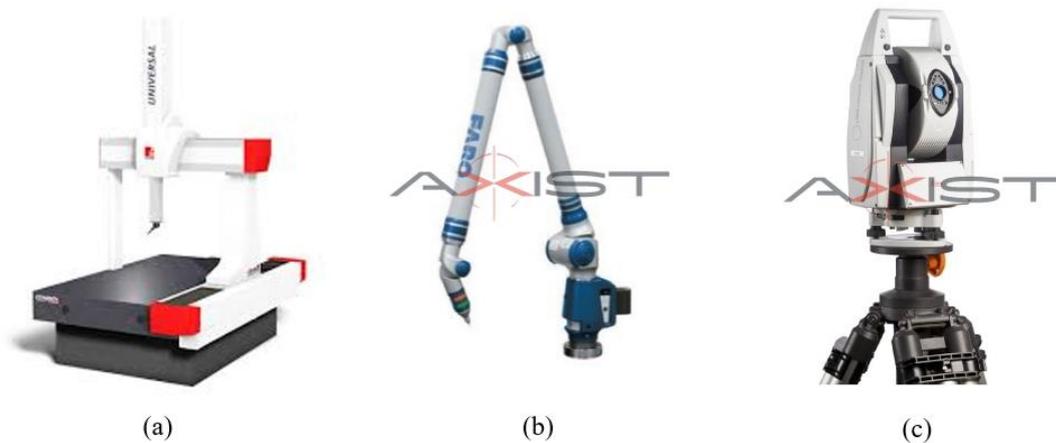


Figura 1.1: (a) CMM Universal della coord3; (b) braccetto di misura Faro, in uso presso Axist; (c) laser tracker Leica AT402 in uso presso Axist

In particolare, le CMM, ovvero le macchine di misura a coordinate, sono ampiamente usate per il controllo qualità ed il reverse engineering. Possono essere suddivise in tre grandi famiglie:

- A portale mobile
- A portale fisso
- A braccio orizzontale

Se le prime due vengono impiegate principalmente in ambienti dedicati al collaudo dimensionale e quindi per controlli campione, l'ultima tipologia a braccio orizzontale si presta bene all'uso in linea all'interno di celle automatizzate.

Per collaudi dimensionali e misurazioni di componenti di notevoli dimensioni, di pezzi unici o per effettuare misurazioni in ambienti al di fuori di quelli canonici di fabbrica, trovano invece largo impiego i sistemi laser tracker ed i braccetti meccanici. In entrambi i casi risulta fondamentale la presenza di uno o più operatori specializzati. A questi sistemi di misurazione vengono solitamente affiancati dei software di misurazione dedicati, che variano a seconda delle case produttrici. Tali software facilitano la programmazione e la simulazione dei piani di scansione consentendo all'operatore di avere maggior controllo e libertà durante il processo di scansione stesso. Tra questi software di collaudo si inseriscono i prodotti SpatialAnalyzer e PolyWorks, entrambi largamente usati dai tecnici e operatori Axist.

In questo contesto dinamico si inserisce Axist S.r.l. che fa da sempre del collaudo dimensionale il suo core-business. Per soddisfare la necessità di una maggiore automazione della misura, richiesta dalle aziende con l'obiettivo di poter effettuare continue ispezioni in linea, Axist ha notevolmente potenziato gli investimenti nel settore di ricerca e sviluppo negli ultimi. In linea con questa filosofia aziendale e grazie anche alla numerosa esperienza sul campo maturata, è stata fondata la divisione Axist Robotics. Tale settore aziendale si occupa dello sviluppo di tecnologie all'avanguardia in ambito metrologico quali, ad esempio, l'applicazione di bracci antropomorfi commerciali e non per collaudi in linea senza l'ausilio di tracciatori esterni. A questo si affianca anche la nuova divisione Automation che realizza applicazioni di automazione industriale basate su controllo Beckhoff, del suo ambiente TwinCAT ed interfacce HMI dedicate. Tra queste applicazioni spiccano strumenti per la misurazione di campi magnetici, tool di misurazione per ispezioni all'interno di giunti e fori e sistemi di visione. (Axist S.r.l, 2020).



Figura 1.2: esempi applicativi di Robot Avris

1.2 Scopo del progetto

Lo sviluppo dell'Avris Rotational Scanning presso il reparto di ricerca e sviluppo di Axist, che verrà indicato all'interno dell'elaborato di tesi anche con l'acronimo "ARS", si basa sull'idea di voler creare un sistema polivalente per la misurazione tridimensionale di oggetti e la determinazione e valutazione, completamente automatiche, dei parametri geometrici. Tale sistema nasce anche dall'esigenza di diversificare l'innovazione tecnologica aziendale che al momento fa del robot Avris il suo progetto cardine. Come punto di partenza, ed in accordo con i target aziendali, si è scelto di indirizzare lo sviluppo verso una misurazione di tipo ottica, tramite la quale i parametri geometrici dell'oggetto da analizzare sono acquisiti senza contatto. I vantaggi nell'utilizzo di un sistema automatizzato ed ottico risiedono nel fatto di eliminare le incertezze dovute all'operatore, tipiche della misurazione classica manuale, e di evitare, nel caso della misurazione a contatto, che la pressione influenzi i risultati ottenuti specialmente se il materiale dell'elemento da analizzare è deformabile.

L'intero processo di misurazione deve essere quindi eseguito in maniera automatica, partendo dal profilo nominale del componente. Per questo motivo è possibile utilizzare la definizione "CAD drive" con cui indicare il contesto applicativo in cui va ad inserirsi il progetto. Un'altra possibile applicazione di notevole interesse risulta essere il reverse engineering dove, a differenza del caso precedente, non è noto il modello matematico del componente, che verrà quindi ricostruito a valle del processo di scansione stesso. In questo caso bisogna affiancare alle funzioni di processo automatizzate una parte hardware e software (sensoristica) atta a garantire la sicurezza in fase di movimentazione e scansione.

I target funzionali che hanno indirizzato lo sviluppo sono:

- Accuratezza della misurazione, conforme agli standard aziendali;
- Ripetibilità della misurazione, fondamentale nel caso di montaggio in linea e misurazione di componenti uguali;
- Velocità di scansione;
- Versatilità di impiego e quindi capacità di adattamento a geometrie diverse;
- Automatizzazione dei processi in modo da ridurre al minimo l'interazione uomo-macchina;
- Integrazione nell'ambiente Avris (Programma sviluppato internamente in Axist).

Se i primi tre punti sono da considerare obiettivi standard per sistemi di misurazione di nuova generazione, gli ultimi due sono dei target che guardano già al possibile uso commerciale del sistema. Automatizzare la maggior parte dei processi e creare delle interfacce semplici si traduce in una maggiore facilità di utilizzo per il cliente. Allo stesso modo, riuscire a creare una suite Avris, dentro la quale possono lavorare e coesistere il sistema ARS ed uno o più robot Avris, permetterebbe di avere un ventaglio esteso di possibili applicazioni.

1.3 Analisi strutturale del sistema ARS

La meccanica dell'Avris Rotational Scanning nasce dalla riconversione di un precedente sistema usato per applicazioni diverse da quelle metrologiche. In questo paragrafo viene quindi presentato ed analizzato il prodotto Gudel TM-40/EP-3 visibile in figura 1.3. Tutte le immagini e le informazioni riportate sono state prese dalla manualistica ufficiale del sistema, fornita direttamente dalla Gudel. (GUDEL, 2003)

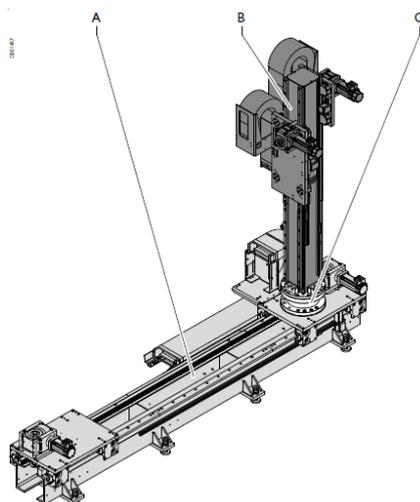


Figura 1.3: Struttura Gudel, prodotto TM-40/EP-3 con opzione C

La struttura è costituita dai seguenti gruppi costruttivi:

- A TM-40
- B EP-3
- C Gruppo rotazione montante C



Figura 1.4: Sistema montato e cablato nel reparto di ricerca e sviluppo

In figura 1.4 invece viene proposta una foto della struttura attualmente posizionata nel reparto di ricerca e sviluppo Axist.

I vari azionamenti presenti nei gruppi consentono, allo stato attuale, di effettuare le movimentazioni lungo gli assi visibili in figura 1.4.

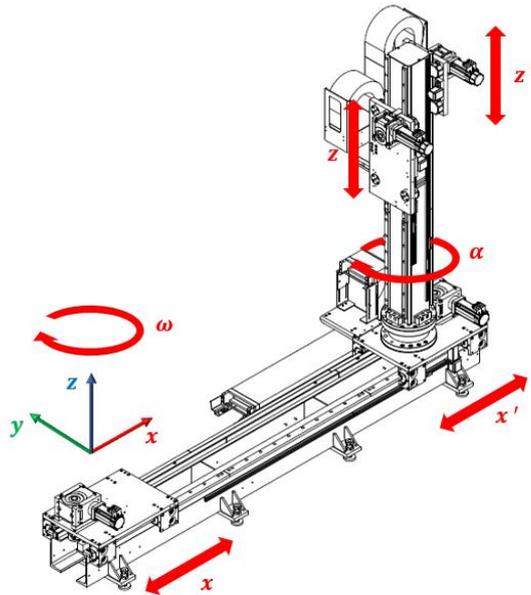


Figura 1.5: sistema di riferimento base adottato e movimentazioni possibili

In funzione dell'applicazione metrologica l'intero sistema può essere visto come l'insieme di due sottosistemi che lavorano in maniera congiunta:

- Gruppo tavola, costituito dall'asse prismatico del carrello che permette la traslazione della tavola e dall'asse rotoidale che permette la rotazione della tavola. Da un punto di vista hardware, il sottogruppo è formato dal prodotto Gudel TM-40 e dalla tavola rotante Newport RV120CC-F;
- Gruppo montante, costituito dall'asse rotoidale " α ", dall'asse prismatico "z" e da un secondo asse rotoidale, non presente nel sistema originario ed in fase di sviluppo, che consentirà di ruotare il tool di misura rispetto l'asse Y del sistema di riferimento base. Da un punto di vista hardware, il sottogruppo è formato dal prodotto Gudel EP-3, dal gruppo opzionale C e dall'asse aggiuntivo per la rotazione del tool.

Sottosistema Tavola

Il prodotto TM-40 è concepito per la movimentazione ed il posizionamento di robot. Nel caso applicativo del sistema ARS, così come per l'uso precedente, viene impiegato per la movimentazione della tavola rotante. L'azionamento che si occupa quindi della movimentazione lungo l'asse X del sistema di riferimento di base è composto da:

- Drive e servomotore kollmorgen;
- Riduttore a vite senza fine con rapporto di trasmissione $i = 24$;
- Ingranamento pignone - cremagliera.

In figura 1.5 (a) è riportato l'ingranamento del pignone (F) con la cremagliera mentre nell'immagine (b) viene riportata la configurazione finale del carrello comprensivo di servomotore montato sull'asse Gudel.

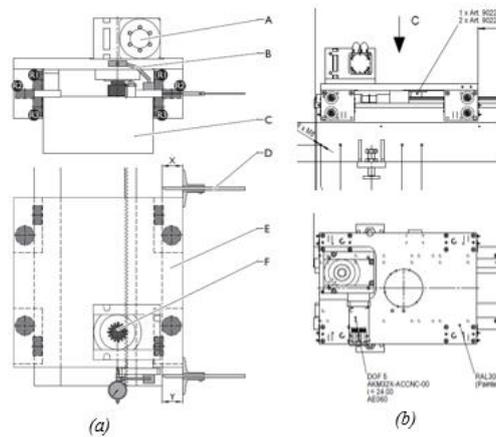


Figura 1.6: asse ed azionamento traslazione tavola rotante

Sul carrello è montata una tavola rotante Newport RV120CC-F, che costituisce il secondo ed ultimo asse del sottogruppo tavola. Questo particolare modello offre un'elevata accuratezza di posizionamento angolare combinata ad un'elevata capacità di carico. Presenta un azionamento a corrente continua ad alta velocità che consente movimentazioni fino a 80°/s mentre la precisione nella rotazione è assicurata da una vite senza fine rettificata e temprata. L'encoder rotativo, che fornisce una risoluzione di 0,001°, è montato direttamente sull'albero dell'ingranaggio a vite a valle della trasmissione a cinghia del motore interno. La rotazione unidirezionale presenta una ripetibilità media pari a 0,001° (0,002° garantita), gli stessi valori si hanno per la ripetibilità in caso di rotazione bidirezionale. La capacità di carico della tavola in condizioni nominali, quindi con carico centrato, è pari a $C_z = 1800 N$. Nel caso di carico non centrato con configurazione ad asse verticale (figura 1.6), si fa riferimento alla seguente formula:

$$Q_H = \frac{C_z}{1 + \frac{D}{a}}$$

Dove D (mm) è la distanza del carico Q_H rispetto all'asse di rotazione mentre a (mm) è un parametro fornito direttamente dal costruttore. (NEWPORT, 2018)

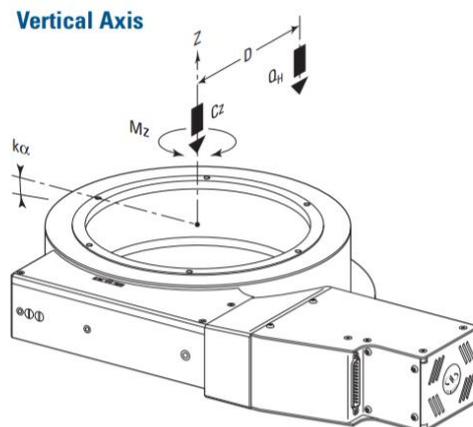


Figura 1.7: sistema di riferimento per il calcolo della capacità di carico

Sottosistema Montante

Il corpo principale è costituito dal prodotto EP-3 che è concepito per la movimentazione e il posizionamento di pezzi e merci. Nel caso applicativo del sistema ARS, dei due carrelli a scorrimento verticale viene sfruttato solo quello posto frontalmente alla tavola rotante (nella condizione di montante non ruotato) che in figura 1.4 corrisponde alla movimentazione Z. In futuro ci si riserva di utilizzare anche il secondo carrello per una possibile attività di machine vision. Alla base del montante è posizionato il gruppo rotazione montante costituito dall'alloggiamento del riduttore planetario, dalle flange di collegamento tra il supporto e montante, figura 1.7.

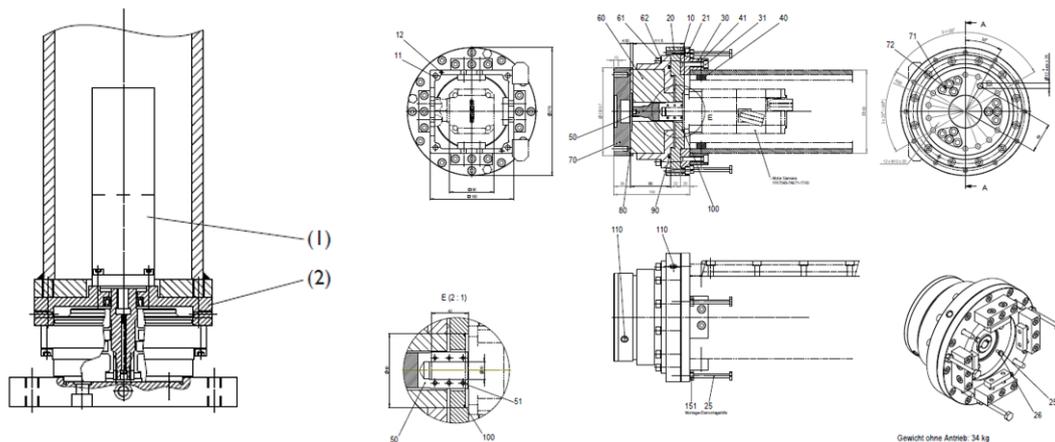


Figura 1.8: Collegamento carrello-montante e alloggiamento riduttore planetario

La scelta di utilizzare un riduttore planetario è dovuta al fatto che il motore dell'azionamento è posizionato all'interno del montante stesso e, grazie a questa particolare tipologia di riduzione, è stato possibile garantire la coassialità tra l'asse in ingresso (asse motore) e l'asse in uscita (asse rotazione montante). In particolare, il riduttore montato, visibile in figura 1.8, è un ingranaggio di precisione Teijin RV-110E-175.28 A-B a due stadi di riduzione, in modo da ridurre vibrazione ed inerzie, per un rapporto di trasmissione complessivo pari a 175.28. (GUDEL, 2003)

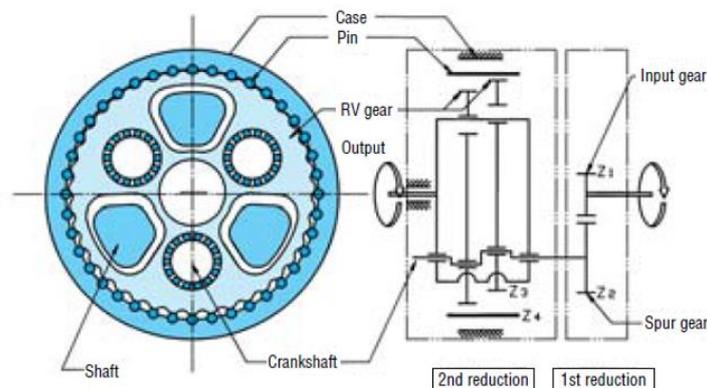


Figura 1.9: schema del meccanismo interno del riduttore Teijin RV-E series

1.4 Quadro elettrico ed architettura hardware di controllo

In prossimità della struttura è posizionato il quadro elettrico al cui interno sono posizionati i seguenti moduli:

- PC Beckhoff
- Moduli Beckhoff
- Drive Kollmorgen
- Adattatori segnale encoder tavola
- Blocco alimentazione
- Switch Ethernet

Il PC industriale Beckhoff è un C6930-0060 progettato per l'installazione in armadio di controllo. L'unità è dotata di una scheda madre da 3½ pollici Intel Core i7 2.9GHz ed un disco fisso di 240 GB capace di garantire elevate prestazioni e velocità di risposta. Le alette di raffreddamento dietro il pannello laterale destro consentono il funzionamento del PC a temperature fino a 55 ° C mentre l'alimentazione richiesta è pari a 24V.

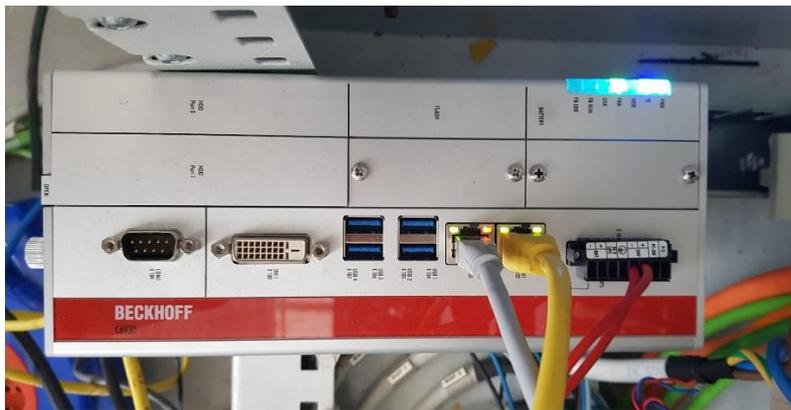


Figura 1.10: PC-Beckhoff C6930-0060

L'interfaccia HW del dispositivo presenta nella porta X101 il Power Supply, nelle porte X102 e X103 i connettori RJ-45 per la connessione in rete, degli ingressi USB nelle porte X104-107, una uscita DVI ed un'interfaccia seriale COM1 nelle porte X108 e X109.

L'architettura Beckhoff è invece composta dai moduli evidenziati in verde in figura 1.10. Partendo da sinistra si ha:

- EK1100, il quale è un accoppiatore tra il fieldbus EtherCAT e gli altri moduli ;
- EL2502-0010, modulo output PWM a 2 uscite;
- EL7342, modulo di controllo per due motori DC;
- EL1809, modulo di digital input a 16 canali;
- EL2809, modulo di digital output a 16 canali;
- EK1110, modulo terminale con estensione della connessione Ethernet.

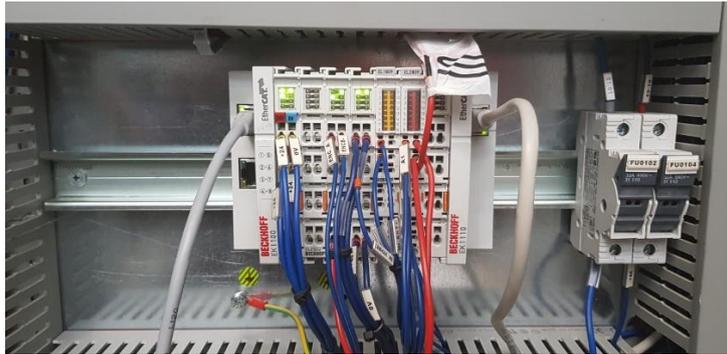


Figura 1.11: terminali Beckhoff

Tramite il modulo terminale si passa direttamente ai sette servo azionamenti Kollmorgen, dei quali sono effettivamente usati i primi due (rotazione montante e traslazione montante) e gli ultimi due (traslazione tavola e traslazione tool). Nei Pin input di ciascun drive arrivano i segnali di Enable e dei finecorsa meccanici, mentre in uscita si avranno tutti i segnali per l'azionamento dei motori



Figura 1.12: Drive Kollmorgen

La rotazione della tavola rotante è gestita dal modulo di controllo Beckhoff collegato all'azionamento Newport. Ciò ha reso indispensabile l'utilizzo di adattatori per il segnale proveniente dall'encoder tavola in quanto il segnale proveniente dalla tavola arriva su quattro canali mentre il modulo EL7342 prevede solamente due ingressi.

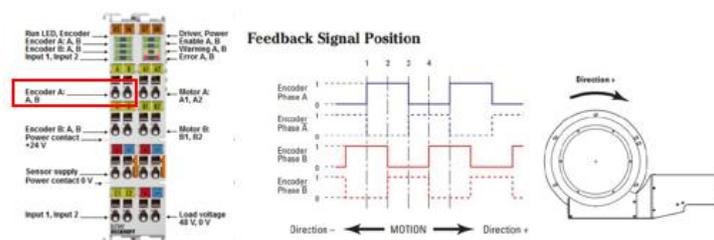


Figura 1.13: Adattatori di segnale dell'encoder rotazione tavola

L'alimentazione a tutti i moduli visti in precedenza viene fornita da due trasformatori AC/DC posti al centro del quadro elettrico e che in ingresso ricevono una tensione che va dai 100 ai 240 Vac restituendo invece 24 e 48 Vdc. Infine, per riuscire a connettere sia il PC-Beckhoff che il tool di misura alla rete aziendale, e quindi controllarlo in remoto, è stato installato uno switch momentaneo nella zona inferiore della cabina.



Figura 1.14: (a) Blocco alimentazione; (b) Switch

1.5 Introduzione ai software usati

Durante l'esecuzione di una routine di scansione del sistema ARS vengono utilizzati quattro diversi software, alcuni dei quali girano in maniera sincrona in varie fasi del processo di scansione:

- SpatialAnalyzer
- Matlab
- TwinCAT
- Avrisc

Durante la fase di prototipazione e sviluppo, sono stati utilizzati altri programmi per risolvere aspetti maggiormente specifici tra i quali: il software di disegno Solidworks per la realizzazione del modello CAD delle guide, il software di misura interno al tool (Gocator LMI), il software Tracker Pilot utilizzato in fase di calibrazione, il software WorkBench della Kollmorgen per il setup degli azionamenti. Nei successivi paragrafi verranno descritti nel dettaglio i quattro programmi principali, figura 1.9, in modo da avere una sezione di riferimento all'interno dell'elaborato da citare nel momento in cui si farà uso di specifiche funzionalità dei vari programmi.

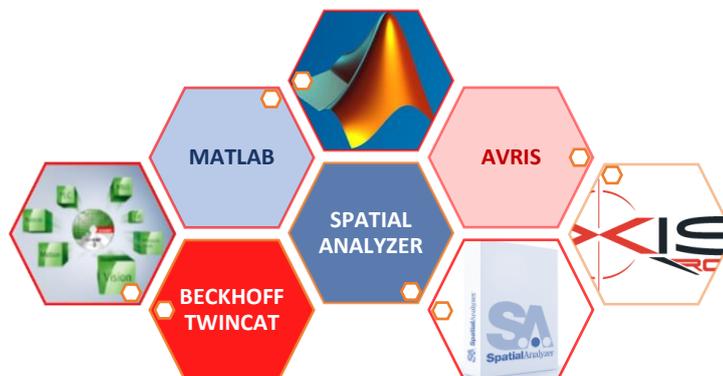


Figura 1.15: software usati durante la routine di scansione

TwinCAT Beckhoff

“The **Windows Control and Automation Technology**” è un ambiente di controllo che consente l’automazione industriale di tipo PC-based, ovvero capace di trasformare ogni PC in un sistema realtime integrando al suo interno l’ambiente di sviluppo. In modo da ottenere le massime prestazioni dal sistema nelle applicazioni industriali, si fa solitamente uso di PC Beckhoff che si basano su sistemi operativi Windows e che sono ottimizzati per il funzionamento realtime (ciò non toglie che qualsiasi PC con TwinCAT può essere utilizzato per l’automazione di sistemi e/o processi). All’interno dello stesso software è quindi possibile integrare la gestione del PLC, della parte di motion ed HMI in modo da ottenere una riduzione della complessità dell’interfaccia, tempi ciclo inferiori, diagnostica migliorata e per ultimo, ma non meno importante, una significativa riduzione dei costi. In generale i sistemi operativi Windows non hanno caratteristiche realtime e quindi non sono stati progettati per gestire task di controllo. La presenza stesso dello scheduler implica infatti che i vari task possono essere interrotti durante la loro esecuzione da una serie di altri eventi con priorità maggiore. Questa tipologia di funzionamento va in contrasto con la normale esecuzione ciclica di un programma PLC che prevede le fasi di lettura input, esecuzione codice e scrittura output. TwinCAT, tramite la sua estensione realtime, consente l’esecuzione sia delle operazioni del sistema operativo che del programma PLC tramite un meccanismo brevettato che consente di restituire il controllo a Windows, figura 1.10.

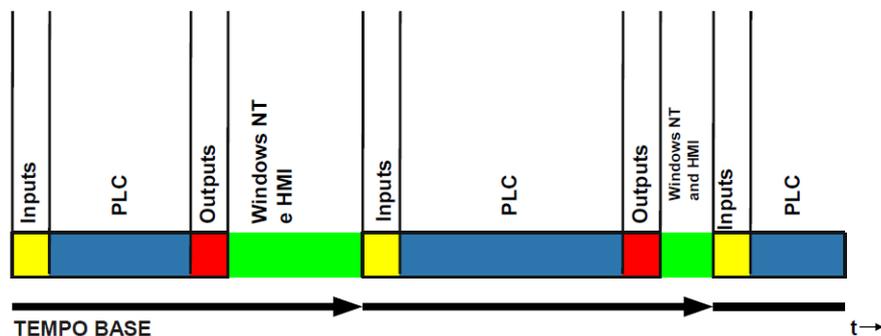


Figura 1.16: scansione TwinCAT PLC

Altro punto di forza della tecnologia di controllo Beckhoff è che all’interno di un unico ambiente è possibile gestire diversi moduli e funzionalità senza dover integrare e quindi investire su nuovi hardware e software. Infatti, all’interno di TwinCAT, oltre alle funzionalità PLC sono integrati moduli per il motion control, per la safety, per l’HMI, per gli I/O, per la connettività, per la programmazione C++ e per la tecnologia di misura. Ciascun modulo è capace di funzionare in modo del tutto indipendente o in parallelo ad altri tramite una comunicazione interna ed esterna via ADS. Nel caso applicativo di progetto, si è fatto largamente uso dei moduli PLC e motion che mettono a disposizione una notevole gamma di librerie e funzionalità già implementate e pronte all’uso. Per quanto riguarda l’interfaccia di programmazione viene seguita la norma IEC 61131-3 che include i seguenti linguaggi:

- Sequential Function Charts (SFC)
- Instruction List (IL)
- Ladder Diagram (LD)

- Function Block Diagram (FBD)
- Structured Text (ST)

L'ultima tipologia, Structured Text, è quella che è stata usata all'interno del progetto ARS, in quanto risulta essere il linguaggio predefinito per tutti i progetti di automazione all'interno della divisione Axist Automation. Allo standard internazionale fanno riferimento anche la struttura del programma (POU), la dichiarazione delle variabili e gli operatori e funzioni di base. Per quanto riguarda le POU (program organization Unit) la normativa prevede tre differenti unità di software:

- Programmi (PRG), possono chiamare blocchi funzioni, funzioni e altri programmi e mantengono memoria dello stato delle variabili interne anche nei cicli PLC successivi.
- Blocchi funzionali (FB), prevedono variabili in ingresso ed in uscita ma è necessario istanziarle. Anche queste mantengono in memoria lo stato delle variabili interne nei cicli PLC successivi.
- Funzioni (FUN), possono avere più variabili in ingresso ma solo una in uscita, non richiedono istanze e non mantengono in memoria lo stato delle variabili interne.

All'interno delle librerie ufficiali sono già presenti una serie di function block e function per la gestione delle operazioni più comuni nei progetti di automazione quali temporizzatori software, trigger, timer, convertitori di variabili, etc. Tendenzialmente, all'interno del programma principale "Main" viene definito il task principale che definisce l'intero ciclo PLC con le annesse chiamate e istanzamenti ad eventuali FB presenti. Per la gestione del ciclo macchina si è fatto invece uso della "macchina a stati", una tecnica di implementazione largamente usate nell'automazioni industriale. Tale programmazione prevede di suddividere le varie fasi di processo e movimentazione in una serie di step, che prendono il nome di stati macchina. Generalmente per entrare nel primo stato macchina di inizializzazione e reset delle variabili devono essere verificate delle condizioni generiche di sicurezza e start. Per passare quindi da uno stato macchina a quello successivo bisogna attendere che la condizione di ingresso sia verificata. In uno stato macchina possono essere associate una serie di azioni, che una volta attivate, permettono al sistema di dare a movimentazioni assi o di passare in particolari configurazioni.

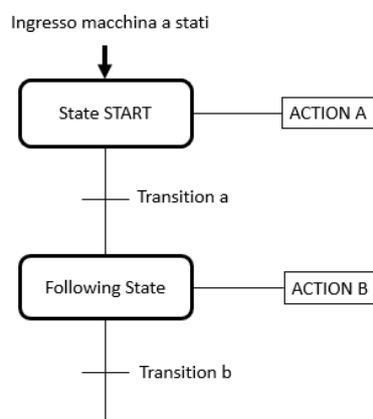


Figura 1.17: schema macchina a stati

Per implementare la macchina a stati in linguaggio ST viene utilizzata la struttura di controllo del flusso “CASE”. All’istruzione CASE viene associata quella che sarà la variabile di stato macchina (di tipo numerico) che a seconda del valore associato permette di entrare all’interno di uno specifico stato. Difatti all’interno del CASE è possibile definire una serie di stati, che corrisponderanno ai vari stati macchina, a ciascuno dei quali viene assegnato uno specifico valore numerico. Tale valore generalmente scala di un fattore cento in modo da consentire l’inserimento di stati, non inizialmente previsti, tra due già esistenti senza andare ad intaccare la sequenza a valle (meccanismo simile alle istruzioni in G-code per macchine utensili). Prendendo come riferimento il codice di figura 1.12, si vede come una volta eseguite le istruzioni dello stato attivo si passa al successivo andando ad assegnare alla variabile di stato il valore numerico corrispondente (state := 1;) se e solo viene verificata la condizione di transizione. Il PLC, infatti, ciclerà all’interno dello stesso stato macchina fino a quando non saranno verificate le condizioni di ingresso.

```

CASE state OF
0:      Q0:=TRUE; Azione
        IF Transition THEN state := 1; END_IF
1:      Q1:=TRUE;
        IF Transition THEN state := 2; END_IF Transizione
2:      Q2:=TRUE;
        IF Transition THEN state := 3; END_IF
3:      Q3:=TRUE;
        IF Transition THEN state := 0; END_IF
END_CASE

```

Figura 1.18: implementazione della macchina a stati in linguaggio ST tramite l’istruzione CASE

Come già accennato in precedenza, TwinCAT mette a disposizione all’interno dello stesso ambiente di sviluppo una serie di moduli, ciascuno dei quali con specifiche funzioni.

Motion

Consente la gestione di diverse tipologie di azionamenti (stepper, servo drive, etc.) consentendo un elevato livello di astrazione in quanto vi è la possibilità di costruire un asse virtuale in parallelo a quello reale. All’interno del pacchetto motion vi sono le seguenti funzionalità:

- NC PTP (Motion punto-punto);
- NC I (Motion interpolato);
- CNC (Motion per macchine a controllo numerico);
- Robotics (Motion interpolato per controllo robot)

Durante lo sviluppo del progetto ARS si è fatto uso della funzionalità NC PTP per i seguenti aspetti:

- Possibilità di gestione di servo azionamenti elettrici;
- Programmazione mediante Function Block per TwinCAT PLC;
- Tempi cicli ridotti, con la possibilità di scendere fino a 50µs;

In particolare, tutti gli assi sono stati configurati come “assi continui” in modo da avere la possibilità di gestire continue variazioni del setpoint in posizione. A ciascun asse è poi possibile associare una configurazione generica, una configurazione encoder ed una configurazione drive in modo da settare i vari parametri di targa motore ed encoder e di scaling (rapporti di trasmissione). In figura 1.13 viene proposto un classico esempio di configurazione preso direttamente dalla manualistica Beckhoff.

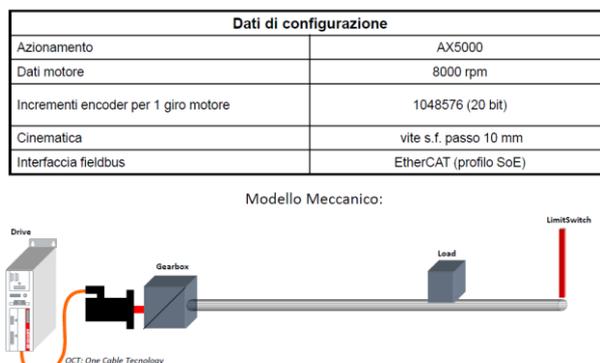


Figura 1.19: configurazione generica di un azionamento Beckhoff

In figura 1.14 si riporta invece la configurazione effettivamente usata nel progetto ARS, dove è possibile distinguere gli assi che caratterizzano il sistema e l'interfaccia di configurazione del generico asse.

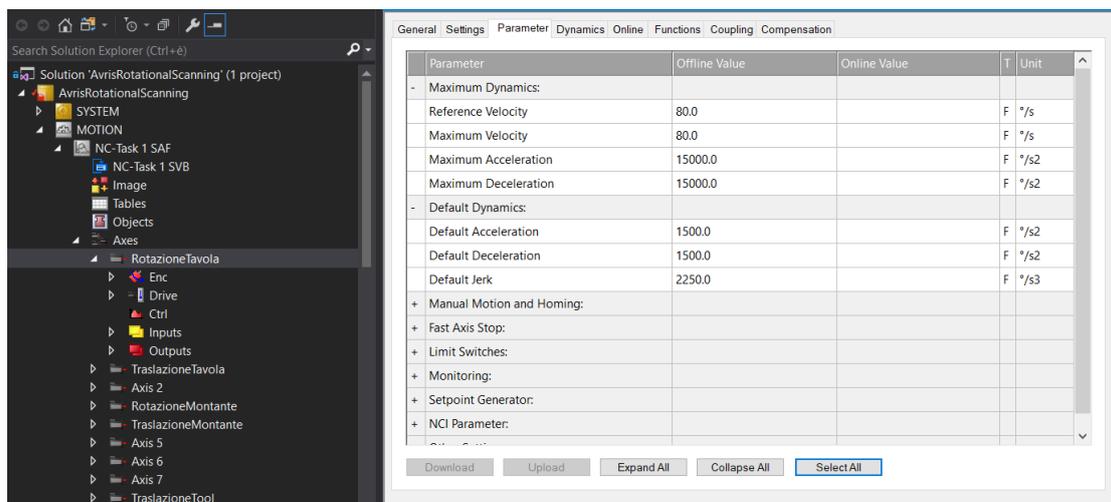


Figura 1.20: Modulo Motion della solution ARS

PLC

All'interno della solution è possibile gestire uno o più progetti PLC per un numero di Task massimo pari a 65000. Una volta creato un progetto PLC, figura 1.15, viene automaticamente creata dal sistema una task standard che chiama il program MAIN e che viene eseguita ogni 10 ms. A questo punto è possibile procedere con l'implementazione del codice, sfruttando le apposite librerie o creando da zero specifici blocchi funzione. Se il progetto PLC viene compilato senza errori, viene generata l'istanza del progetto PLC con all'interno tutte le immagini di processo SW. Ciò consente al modulo PLC di interagire con gli altri moduli SW presenti e di collegarsi alle immagini di processo HW.

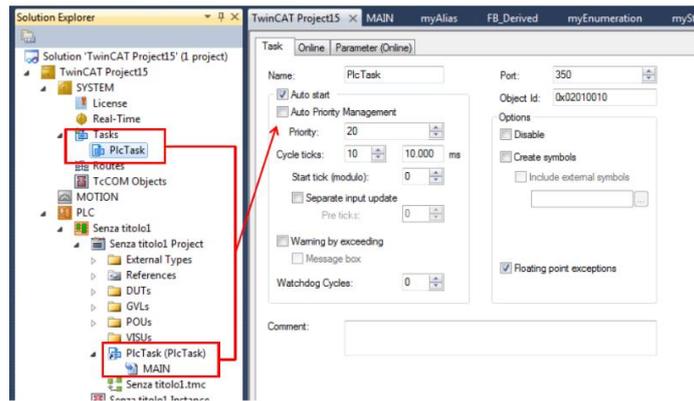


Figura 1.21: task principale del progetto PLC

Vengono infine riportati in figura 1.16 alcuni passaggi presi dal codice PLC implementato per il sistema ARS, in modo da meglio comprendere la logica di dichiarazione delle variabili (a), dell'istanziamento delle function block (a), della macchina a stati (b)

<pre> 1 PROGRAM MAIN 2 VAR 3 fb_adsIO : fbAdsIO; 4 //trig HMI 5 trgStart : R_TRIG; 6 trgReset : R_TRIG; 7 trgStop : R_TRIG; 8 trgRestart : R_TRIG; 9 //trig ADS 10 trgAdsTfsetted : R_TRIG; 11 trgMoveDoneAds : R_TRIG; 12 trgArvrisDoneAds : R_TRIG; 13 END_VAR 14 15 _Input(); 16 fb_adsIO(); 17 _Cycle(); 18 _Axes(); 19 _RequestNewTargetPosition(); 20 _Reset(); 21 _Output(); </pre>	<pre> 1 2 3 IF AutomaticMotion AND trgStart.Q AND NOT AutoRunBusy THEN 4 AutoRunBusy := TRUE; 5 lastState := STATE_START; //entro nel case (inizis 6 State := STATE_START; 7 END_IF 8 9 CASE State OF 10 STATE_START: (* Reset variabili *) 11 //bEnableAsse := FALSE; 12 bExecute_Move := FALSE; 13 bHomeDone := FALSE; 14 ScansioneDone := FALSE; 15 //TargetPosSetted := FALSE; 16 bExecute_Reset := TRUE; 17 tmr_Reset := Time_out_InitialReset; 18 lastState := State; 19 State := STATE_FIRST_100; 20 21 STATE_FIRST_100: (* Attesa Reset e Abilitazione asse *) 22 IF tmr_Reset = 0 THEN 23 bExecute_Reset := FALSE; 24 bEnableAsse := TRUE; //metto a true i Dig 25 State := 105; 26 END_IF </pre>
(a)	(b)

Figura 1.22: (a) sopra l'editor per dichiarare le variabili, sotto istanziamento di FB e richiamo azioni.
(b) Stati iniziali del ciclo macchina implementato

Comunicazione

In parallelo all'implementazione SW del codice vi è infatti la possibilità di connettersi ad una serie di moduli esterni (Beckhoff e non) per la gestione di I/O, servo azionamenti, sensori, etc. Per far si utilizza il bus di campo industriale EtherCAT che è una rete del tipo master-slave: l'unico nodo che può inviare frame alla rete è il master, gli slave si appoggiano a quest'ultimi per scambiare i propri dati. A prescindere della tipologia di rete implementata l'anello logico che viene seguito è il seguente:

1. Il master EtherCAT invia il frame;
2. Lo slave riceve il frame dal nodo precedente, legge i dati ad esso riferito, scrive in uscita i suoi dati processati e lo invia al nodo successivo;
3. Il frame torna al master dopo aver attraversato tutti i nodi slave.

Per quanto riguarda invece la comunicazione tra moduli software diversi, sia che essi stiano sullo stesso hardware o su due dispositivi distinti, viene utilizzato un protocollo chiamato ADS: Automation Device Specification.

Tale protocollo utilizza una logica del tipo client-server, dove l'ADS Server è generalmente costituito da TwinCAT mentre L'ADS Client è un generico applicativo Windows, anche esterno dall'ambiente di sviluppo TwinCAT, che accede tramite ADS ai dati real-time. Nel progetto di tesi viene utilizzata questa tipologia di comunicazione tramite connessione TCP/IP dato che ADS Client e ADS Server risiedono su macchine diverse. Una generica comunicazione ADS si sviluppa seguendo lo schema di figura 1.17. (Beckhoff, 2020)

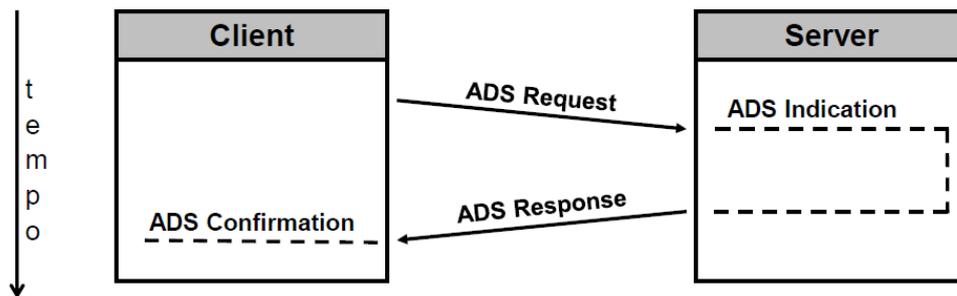


Figura 1.23: comunicazione ADS tra Server e Client

SpatialAnalyzer

SpatialAnalyzer (SA), sviluppato e distribuito dalla New River Kinematics, è un software per strumenti metrologici portatili e applicazioni su grandi volumi. SA è una piattaforma grafica 3D che consente l'analisi dei dati, la messa a punto, il controllo, l'automazione, la reportistica e il reverse engineering. Il suo utilizzo in questo progetto di tesi nasce dal fatto che, assieme a PolyWorks della Innovmetric, è il principale software di misura attualmente in uso in Axist. Tra le caratteristiche principali vi è la possibilità di associare una vasta gamma di strumenti di misura commerciali o di implementare uno strumento di misura custom (con modello matematico e cinematica annessa). SA permette il collaudo dimensionale con le tolleranze secondo lo standard ASME GD&T e di definire delle routines partendo dalle entità geometriche presenti sul CAD dell'oggetto da misurare. Tra le maggiori affinità con il progetto di tesi vi sono sicuramente le caratteristiche per quanto riguarda l'automazione di alcuni processi. Infatti, attraverso l'utilizzo di piani di misura, detti MP, è possibile implementare script di automazione di misura e di seguire in tempo reale le attività dello strumento di misura confrontando la nuvola di punti misurata con le entità e geometrie nominale del modello matematico.

Generalmente una suite di SA è composto da tre elementi principali:

- la macchina con in esecuzione il software
- l'interfaccia dello strumento
- lo strumento di misura



Figura 1.24: configurazione generica di un sistema di misurazione con SA

In figura 1.18 viene proposta l'interfaccia standard di SA che si apre non appena si fa partire il programma. All'interno dell'albero di sinistra (SA TreeBar) sarà quindi presente una collezione con all'interno la sola frame del sistema di riferimento base che è possibile vedere nell'area grafica centrale. Sulla tool bar superiore sono presenti una serie di menu a tendina che racchiudono tutte le funzionalità presenti in SA. In particolare, facendo sempre riferimento alla figura 1.19 si ha che:

- sotto Construct (1), vi è la possibilità di definire entità geometriche (punti, linee, cerchi, piani, etc.) partendo sia dai valori misurato che nominali;
- Sotto Instrument (2), vi è la possibilità di aggiungere uno strumento di misura commerciale. All'interno di SA è già preinserita una lista con la maggior parte degli strumenti attualmente presenti sul mercato;
- Sotto Scripts (3), vi è la possibilità di creare un nuovo piano di misura o di caricare e modificare un MP già esistente;
- Sotto Reports (4), sono raccolte tutte le azioni di reportistica;

Per quanto riguarda le Query, le Relationships, le Analysis e le GD&T, SpatialAnalyzer mette a disposizione un apposito toolkit (5) dedicato che ne facilita l'utilizzo.

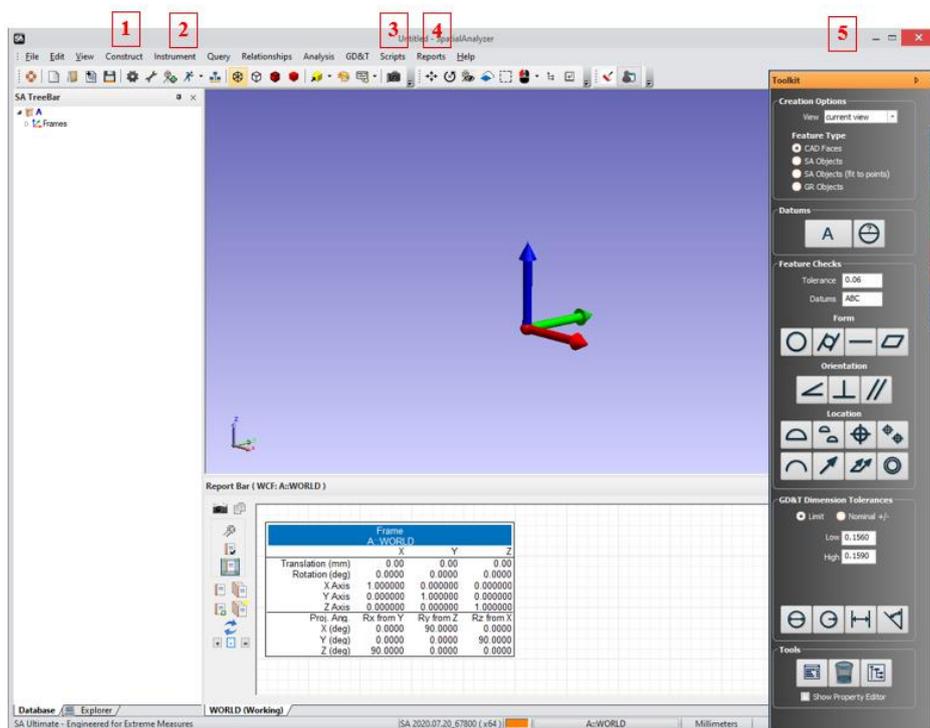


Figura 1.25: interfaccia di SA

Come già accennato in precedenza, una delle funzionalità più utili messe a disposizione dall'ambiente SA è la possibilità di automatizzare una serie di comandi e operazioni tramite l'interfaccia di scripting integrata che prende il nome di "Measurement Plans" (MPs). All'interno degli script è possibile effettuare calcoli, scrivere e leggere file esterni, far partire delle sub routine o degli script esterni tramite SDK. In figura 1.19 viene proposto l'editor MP che fornisce un'interfaccia grafica per l'implementazione del codice. Anche se presenta alcuni tratti di un IDE (Integrated Development Environment) tradizionale, non è un editor di testo ed il linguaggio di programmazione risulta semplificato in quanto costituito da una serie di comandi preconfigurati. In ogni comando è infatti presente una lista degli input richiesti per l'esecuzione del comando stesso e la tipologia di output che si avrà in uscita.

Ci sono tre sezioni principali nell'editor figura 1.20 :

- (A) Command List, che permette di cercare i vari comandi disponibile tramite l'uso di poche keywords;
- (B) Variable View, che permette di visualizzare tutte le variabile usate all'interno del codice;
- (C) Step List, che costituisce la maggior parte della finestra, è che permette di visualizzare l'elenco dei comandi nell'ordine di esecuzione e di modificare gli argomenti.

Per quanto riguarda la convenzione usata da SA si ha che il primo step dello script sarà il passaggio zero, così come sarà zero il primo argomento di un comando o l'indice del primo valore di un array.

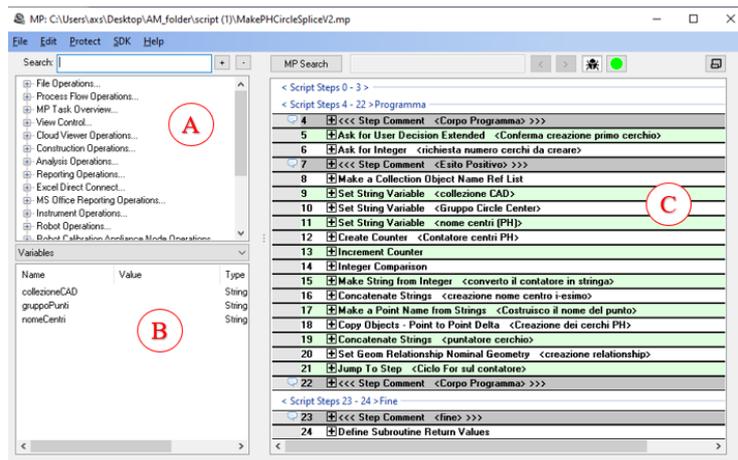
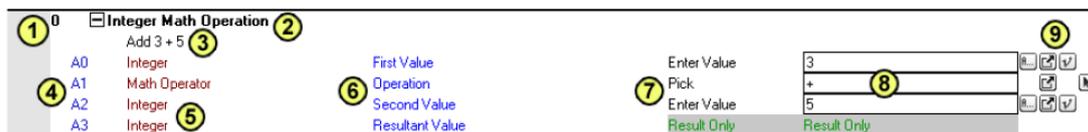


Figura 1.26: MP's Editor

In figura 1.21 viene proposta la forma espansa di un comando, nello specifico il comando per operazioni algebriche su variabili di tipo Integer, appena aggiunto all'interno della Step List.



Viene nel seguito analizzato nel dettaglio un generico comando:

- (1) Step number che identifica in maniera univoca un comando;
- (2) Nome del comando;
- (3) Commento opzionale;
- (4) Argomenti di input e output previsti dal comando;
- (5) Data Type dei vari argomenti;
- (6) Descrizione aggiuntiva dell'argomento;
- (7) Metodi di inizializzazione dell'argomento;
- (8) Valore argomento;
- (9) Metodi di inizializzazione (Quick select).

Tra le altre funzionalità di SA utilizzate durante lo sviluppo del progetto di tesi vi sono l'interfaccia per l'inserimento di una cinematica custom, l'interfaccia dedicata al laser tracker per effettuare la calibrazione (entrambe verranno analizzate nel dettaglio nel Capitolo 3) e la possibilità di configurare tasti e toolbar e salvare il tutto in una suite personalizzata (suite ARS). (Spatial Analyzer, 2020)

2. Logica di sistema

Il sistema ARS sfrutta quattro diversi software per poter effettuare la scansione automatica del componente da misurare:

- SpatialAnalyzer (SA), software di collaudo dimensionale che, oltre alla creazione del piano di controllo e misura, viene anche usato nella fase preliminare di creazione delle traiettorie assi;
- Matlab (ML), ambiente di calcolo numerico utilizzato per il calcolo delle posizioni target;
- Avris, programma sviluppato internamente che consente una gestione completa del sistema e del tool di misura;
- TwinCAT (TC), software che permette di avere, su un classico PC, una suite completa per il controllo e l'automazione del sistema (PLC, NC, CNC).

In figura 2.1 è schematizzata l'intera logica di sistema mentre nei successivi paragrafi vengono analizzate nel dettaglio le varie fasi del processo di elaborazione e scansione.

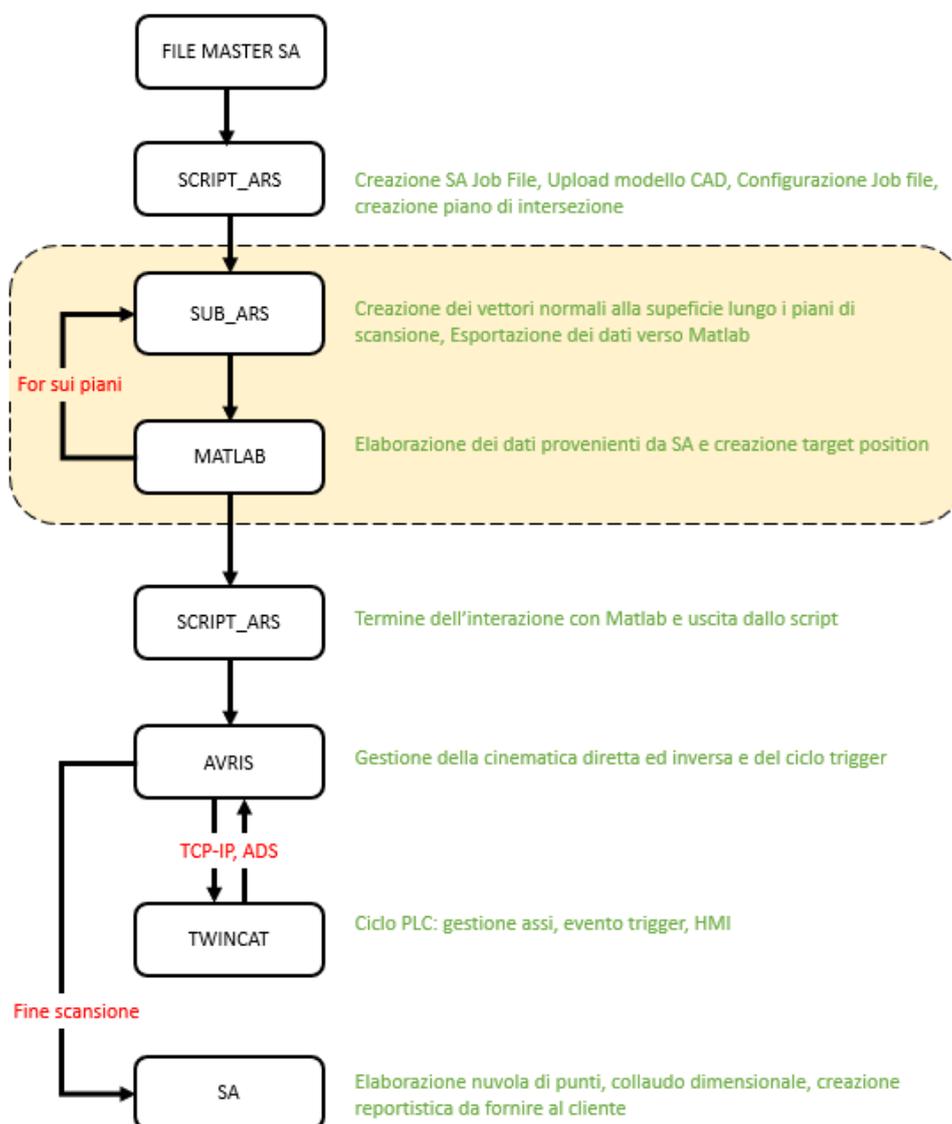


Figura 2.1: processo di scansione del sistema Avris Rotational Scanning

2.1 Configurazione file SA e fase preliminare di calcolo delle posizioni target

Il processo di scansione inizia con l'apertura del file MasterARS_WorkFolder.xit64 contenuto nella cartella di lavoro WorkFolder_ARS. Nel momento dell'apertura, all'interno del file master sarà presente il solo modello CAD del sistema semplificato. Per lanciare lo script principale occorre premere l'apposito tasto "script ARS" presente nella toolbar superiore della Suite ARS appositamente creata per questo progetto, figura 2.2.

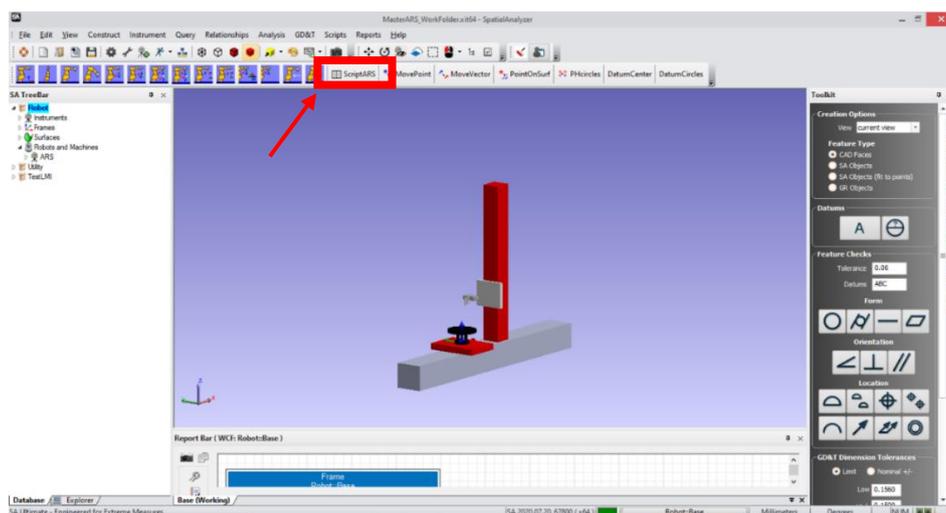


Figura 2.2: file master aperto dal Workfolder e suite ARS con tasto di avvio script

Attivata l'esecuzione dello script, viene eseguito il primo blocco di istruzioni che consente di:

1. Cancellare eventuali cartelle temporanee e, se presente, il file SADone.txt nella cartella di lavoro (ciò permette di mandare in esecuzione parallela anche lo script di Matlab);
2. Creare un nuovo file SA, sempre all'interno del WorkFolder, andando a copiare il file master precedentemente aperto. Questo nuovo file sarà l'effettivo ambiente di lavoro e ciò consente di lavorare senza avere il rischio di corrompere il file master che può essere riaperto in qualsiasi momento in caso di bisogno.

All'operatore è data l'opportunità di rinominare il nuovo file in modo da poterlo identificare in maniera più agevole in fase di analisi dati oppure in modo da inserire il codice del part number del componente da misurare se richiesto dal cliente, figura 2.3.

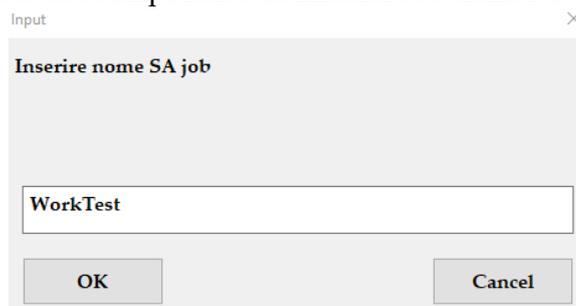


Figura 2.3: Interfaccia utente dello script di SA, rename file

Definito quindi quello che sarà il file di lavoro, lo script prosegue nell'esecuzione del secondo blocco di codice. Esso verifica l'effettiva presenza del modello matematico del componente da misurare all'interno della cartella di lavoro. Il file CAD deve essere quindi posizionato all'interno della cartella di lavoro prima di iniziare la procedura e rinominato "3DModel.Stp" in modo da essere correttamente identificato ed importato all'interno di SA in maniera automatica. Una volta importato il modello nell'ambiente di SA, tramite la GUI creata all'interno dello script stesso, viene chiesto all'operatore di effettuare una verifica sulla corretta posizione del modello matematico del componente (comprensivo di sistema di fissaggio da collegare alla tavola rotante). Accertata la corretta posizione del pezzo sulla tavola, premendo sul comando di start, figura 2.4 (a), si procede con la configurazione dei seguenti parametri richiesti:

- Altezza pezzo da scansionare;
- Quota iniziale pezzo (tiene conto del sistema di fissaggio);
- Percentuale di overlapping della scansione.

In figura 2.4 (b) viene riportata la schermata di recap che appare all'operatore una volta inseriti tutti i parametri richiesti per poter procedere con l'elaborazione dello script.



Figura 2.4: (a) Interfaccia utente dello script di SA per la verifica del posizionamento componente. (b) Interfaccia utente dello script di SA, finestra di recap

Inseriti quindi tutti i parametri richiesti dallo script SA, può iniziare la fase automatica di generazione dei dati da esportare verso Matlab, tramite file ASCII, per la creazione della movimentazione di scansione. L'intero processo è implementato all'interno di un ciclo for che può essere descritto tramite i seguenti passi salienti:

1. All'inizio del ciclo for viene creato un piano, che prende il nome di "piano di intersezione", in prossimità del piano di appoggio della tavola rotante, avente normale lungo Z e alla quota impostata precedentemente dall'operatore. Tale valore deve corrispondere alla quota di inizio materiale del modello da scansionare;
2. Successivamente, il piano viene traslato lungo Z di una quantità pari al valore di overlapping. Con percentuale di overlapping si intende la quota parte del trapezio di scansione che viene nuovamente intercettata dal fascio laser durante la scansione del piano successivo, figura 2.5. Inserendo quindi un valore di overlapping del 25% permette di avere scansionare nuovamente un quarto della superficie precedente. Maggiore sarà l'aliquota selezionata e più sarà fitta la nuvola di punti. Allo stesso tempo, a parità di altezza del pezzo, si avrà un numero maggiore di piani scansioni e quindi un tempo ciclo globale di

scansione maggiore. La scelta del parametro di overlapping dovrà quindi essere ponderata in funzione delle due precedenti considerazioni ed in base alle esigenze del cliente;

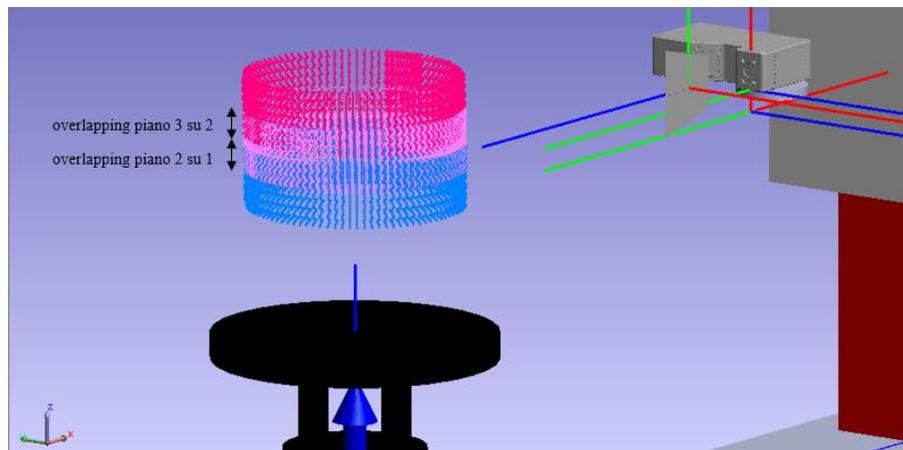


Figura 2.5: esempio di scansione con overlapping

3. L'intersezione tra piano e componente permette di ottenere una serie di Bspline, ovvero una serie di curve che seguono il profilo dell'oggetto lungo la linea di intersezione. Dopo aver unificato le singole B-Spline, a valle della fase di intersezione è presente un comando di verifica che consente di uscire dal ciclo for nel caso di mancata intersezione. Ciò si traduce nel fatto che il nuovo piano shiftato ha superato il pezzo e quindi, tramite un jump, si viene rimandati nel blocco per l'uscita dal codice;
4. A questo punto dello script, si entra nella subroutine "sub_ars" che presenta tra i vari parametri di ingresso la BSpline appena generata. All'interno della subroutine vengono:
 - puliti i file ASCII di scambio dati;
 - creati i vettori uscenti alla superficie del pezzo lungo la bspline;
 - esportati i dati relativi ai vettori sui file di scambio;
 - creato il file per l'autorun dello script Matlab.

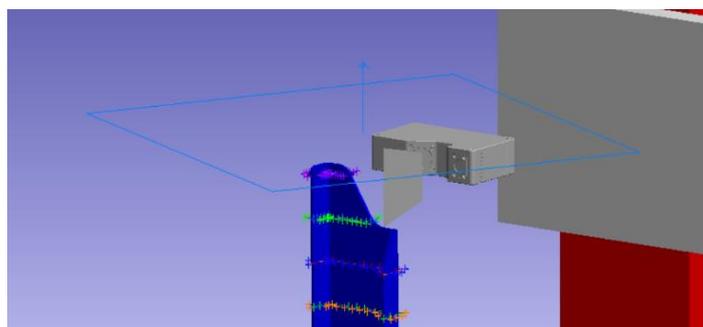


Figura 2.6: Piano di intersezione e vettori normali uscenti creati sui vari livelli di scansione al termine dello script ARS

5. Dopo l'ultima operazione della subroutine, prima di uscire e riprendere il ciclo for per shiftare nuovamente il piano alla successiva quota, si resta in attesa del file di testo "MatlabDone" e si procede con la cancellazione di tutte le cartelle temporanee.

Terminata l'esecuzione dello script è possibile visionare i vettori all'interno dell'ambiente SA e accedere ai singoli punti dall'albero di sinistra sotto la cartella CAD\Point Groups o CAD\Lines. Ciò permette all'operatore di effettuare una rapida verifica dei risultati ottenuti qualora sussistano delle problematiche legate perlopiù a geometrie particolari del componente da misurare.

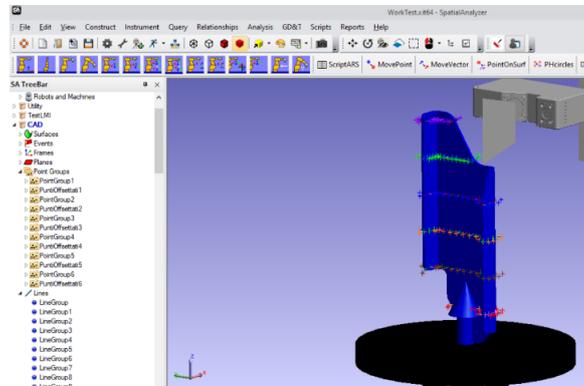


Figura 2.7: Albero di SA una volta terminata l'esecuzione dello script ARS

Il processo descritto precedentemente, per una più completa comprensione, viene riportato graficamente tramite lo schema a blocchi di figura 2.8.

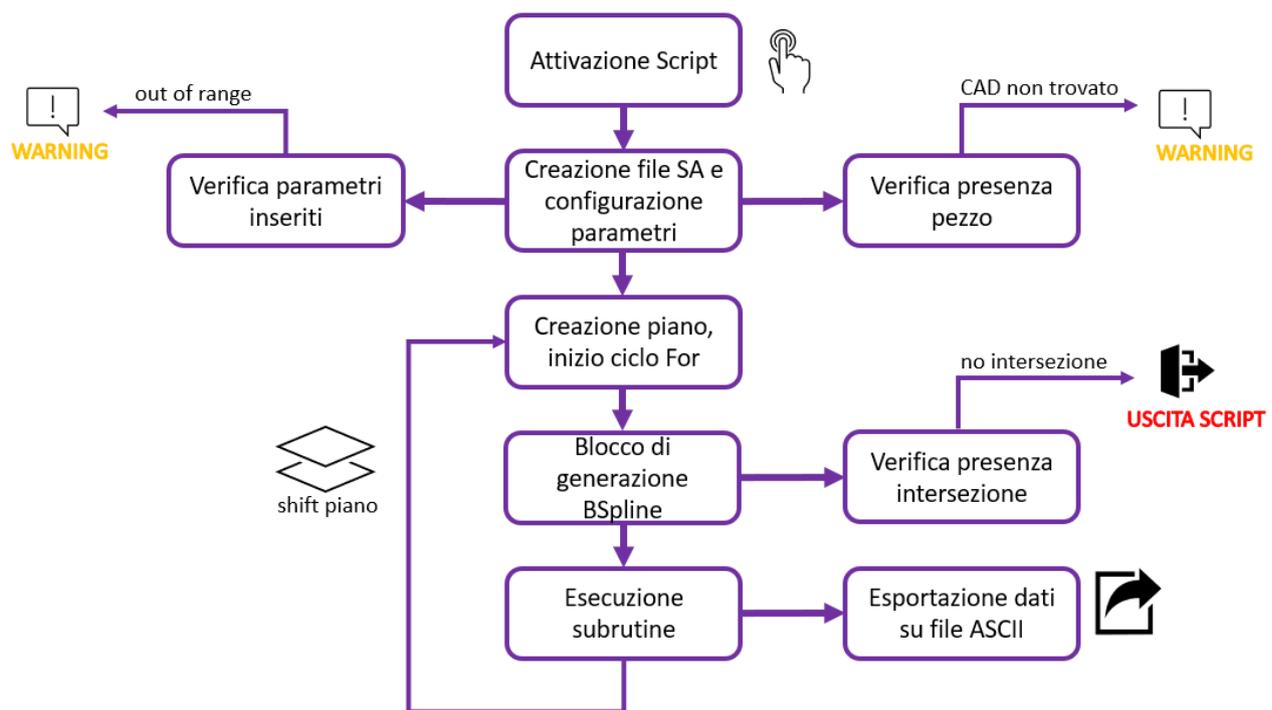


Figura 2.8: Diagramma di flusso dello "Script_ARs"

All'interno dello script principale sono stati inseriti delle task di controllo che assicurano la corretta esecuzione del programma e che affiancano l'operatore in modo da ottenere una corretta configurazione dei parametri. Tra le varie task di controllo, ve ne sono alcune che arrestano il programma nel caso in cui non vengono trovati determinati file o cartelle, come visibile in figura 2.9, oppure altre che mandano a

schermo dei warning nel caso di inserimenti da parte dell'operatore erronei o al di fuori dei range prestabiliti.

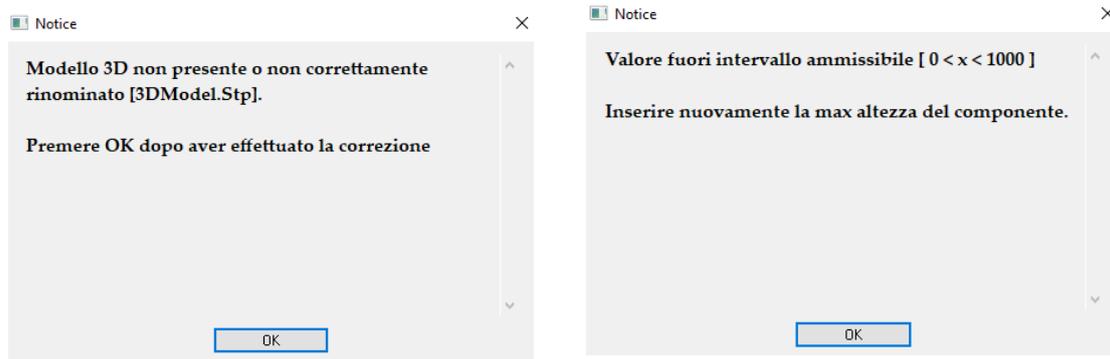


Figura 2.9 Arresto esecuzione del codice per assenza del CAD nella cartella di lavoro e warning causa sfioramento range

2.2 Matlab: calcolo delle posizioni target

Matlab è il software scelto per effettuare il calcolo delle posizioni target da passare successivamente al controllo (TwinCAT-Avril) in modo che, di volta in volta, tutti gli assi del sistema raggiungano le posizioni richieste nel modo corretto.

Per ogni piano di scansione, vengono creati da SA due file ASCII al cui interno sono presenti tutte le informazioni relative ai vettori uscenti calcolati proprio lungo la superficie di intersezione piano-solido. Il vettore viene univocamente definito dalle coordinate del punto iniziale (posizionato sulle superficie del componente) e dalle coordinate del punto finale. I valori relativi ai due punti andranno a costituire tutti gli input diretti a Matlab e provenienti da SA.

La comunicazione con SA e la fase di elaborazione avviene tramite uno script e due function:

- ScambioDati_SA_ML.m
- MMR.m
- SMF.m

2.2.1 ScambioDati_SA_ML.m

Costituisce lo script principale all'interno del quale vengono richiamate a cascata le altre due function, "MMR.m" e "SMF.m", che si occupano della vera e propria elaborazione dati e costruzione delle posizioni target. Tale script viene mandato in Run subito dopo aver mandato in esecuzione lo script interno a SA descritto nel paragrafo precedente. I due script sono quindi stati implementati per lavorare in maniera del tutto sincrona tramite un particolare protocollo di comunicazione che verrà presentato più avanti. Per poter mantenere in esecuzione lo script principale è stato usato un ciclo while la cui condizione di uscita si attiva solo quando sono state elaborate le target position relative all'ultimo piano di scansione. Una volta entrati all'interno del ciclo while è possibile accedere alla prima function "MMR.m" se e solo viene verificato l'IF di ingresso relativo alla presenza del file "STARTMATLAB.txt" e alla contemporanea assenza del file "MatlabDone.txt" nella cartella di scambio dati. Terminata l'esecuzione delle due function interne si rientra all'interno del ciclo while e viene creato il file di testo "MatlabDone.txt", che riconosciuto da SA, permette di avanzare al piano di scansione successivo. Prima di attivare nuovamente le due function di calcolo, si verifica la condizione di uscita: nel caso in cui il file "SADone.txt" sia presente all'interno della cartella di scambio dati l'esecuzione dello script termina istantaneamente. Tale script si rende quindi necessario per effettuare in maniera corretta lo scambio dati fra i primi due software in gioco, ovvero SA e Matlab. Nei paragrafi successivi verrà descritto il tutto in maniera più specifica.

2.2.2 MMR.m

La function "Main Matrix Rotated" viene richiamata direttamente dallo script principale e presenta come input l'ultima posizione raggiunta dalla tavola nel piano di scansione $i-1$ e restituisce come output la posizione finale raggiunta dalla tavola nel piano di scansione i -esimo. Tale output diventerà l'input nella fase di scansione successiva. Per il primo piano scansione, non essendoci ancora una precedente posizione raggiunta, viene preso come valore angolare quello di inizializzazione ovvero

zero gradi (la tavola non ruota durante la prima movimentazione). In figura 2.10, è schematizzato l'intero algoritmo implementato.

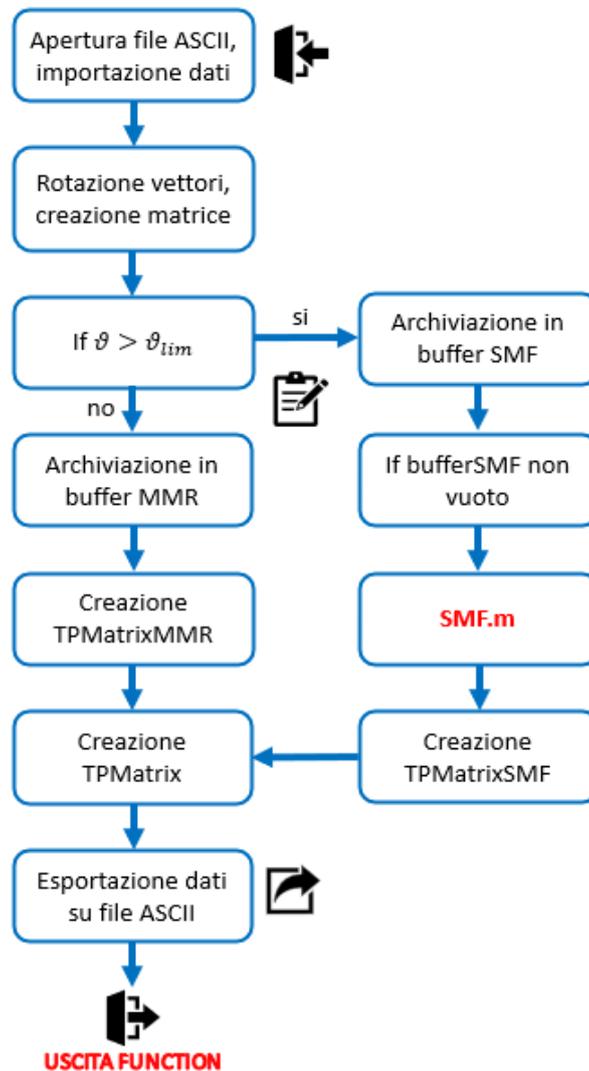


Figura 2.10: Diagramma di flusso function MMR.m

Dopo aver estratto i dati relativi ai vettori uscenti alla superficie lungo il piano di intersezione dai due file ASCII, ha inizio la procedura di determinazione delle posizioni target che verranno poi trasferite alla parte di controllo macchina. Come primo step, si applica a ciascun vettore una matrice di rotazione in modo da portare il punto iniziale del vettore (1) lungo l'asse x (2), figura 2.11. Tale asse corrisponde a quello passante fra i centri di rotazione della tavola rotante e del montante ed anche all'asse del fascio laser quando la rotazione del monetante è nulla.

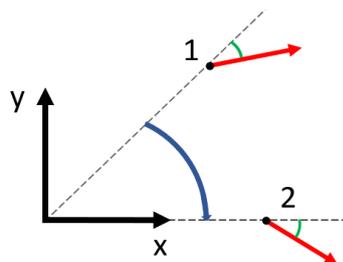


Figura 2.11: Applicazione della matrice di rotazione sul generico vettore

Questo procedimento consente di simulare la rotazione della tavola e di valutare distanze e angoli di incidenza, ovvero gli angoli tra vettore e fascio laser, proprio quando il punto di applicazione del primo viene intercettato dal trapezio di scansione. Creata quindi la matrice dei vettori ruotati relativa al generico piano, si procede allo smistamento dei vettori in due buffer separati:

- `bufferMMR`, nel quale vengono archiviati tutti i vettori che non richiedono alcuna correzione dell'angolo α del montante;
- `bufferSMF`, nel quale vengono archiviati i vettori che presentano un angolo non acquisibile dal trapezio di scansione nella sua posizione nominale e che quindi richiedono una movimentazione del sottosistema tavola-montante.

Dal `bufferMMR` è poi possibile calcolare le target position necessarie a scansionare l'intero profilo del pezzo. Si fa notare che in funzione della geometria del componente potrebbe essere necessario effettuare più scansioni dello stesso piano. Tutte le movimentazioni calcolate vengono quindi archiviate all'interno di una apposita matrice `TPMatrixMMR`. Infine, nella parte terminale dell'algoritmo, si procede ad accodare alla `TPMatrixMMR` le eventuali target position provenienti dalla function `SMF.m` in modo da ottenere la `TPMatrix` del piano i -esimo che sarà poi esportata sul file `TargetPosition.txt`.

2.2.3 SMF.m

L'acronimo `SMF` sta per "Sort Merge Filtered" e rappresenta l'algoritmo più interno di tutto il blocco di calcolo implementato su Matlab. La function "`SMF.m`", che permette di calcolare l'eventuale rotazione del montante, viene richiamato all'interno della function `MMR.m` se e solo se il `bufferSMF`, creato all'interno della stessa "`MMR.m`", non è vuoto. Presenta come input l'ultima posizione della tavola nel piano di scansione i -esimo (posizione archiviata all'interno della matrice `TPMatrixMMR`) e restituisce come output principale la matrice `TPMatrixSMF`, la quale verrà accodata alla corrispettiva calcolata in `MMR` in modo da formare la matrice delle posizioni finali. Gli output comprendono anche dei parametri di controllo, tra i quali troviamo le correzioni da dare alla tavola, lungo l'asse x , a seguito della movimentazione del montante. L'algoritmo implementato all'interno di questa function verrà spiegato più nel dettaglio nel capitolo relativo alla movimentazione del montante.

2.3 Protocollo per lo scambio dati tra SA e Matlab

Tale protocollo nasce dall'esigenza di inviare a Matlab i pacchetti dati che vengono generati da SA per ogni singolo piano di scansione in modo da assicurare un funzionamento sincrono dei due programmi. Tale protocollo sfrutta le seguenti funzionalità proprie dei software:

- Tramite alcuni comandi presenti in SA è possibile creare file ASCII, esportare dati, verificare l'esistenza o meno di determinati file e/o cartelle e cancellare specifici file.
- Tramite alcuni comandi presenti su Matlab è possibile restare in ascolto, leggere e scrivere file, creare e cancellare file e importare ed esportare dati su file di testo.

In figura 2.12, è schematizzato l'intero protocollo di comunicazione implementato che verrà descritto nel seguito.

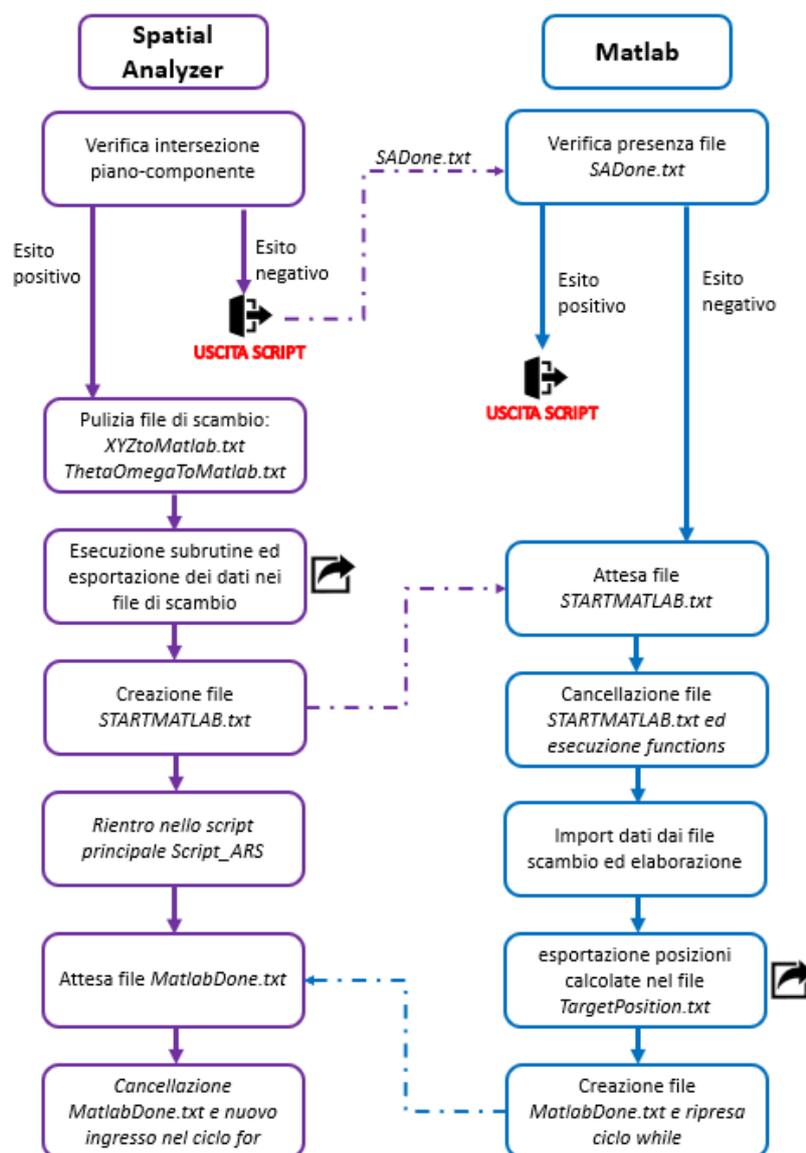


Figura 2.12: Protocollo di comunicazione tra SA e Matlab

Lato SA

All'inizio dell'esecuzione della subroutine "sub_ars" si procede con la pulizia dei file di archiviazione dati *XYZToMatlab.txt* e *ThetaOmegaToMatlab.txt* che sono già presenti nella cartella "ScambioDati" dell'ambiente di lavoro "WorkFolder_ARS". Tale pulizia si rende necessaria onde evitare che la presenza di informazioni e dati relativi a piani di scansioni precedenti possa interferire con il calcolo corrente. Effettuato il calcolo ed esportati i dati sui file di testo precedentemente puliti, viene creato un nuovo file di testo *STARTMATLAB.txt* all'interno sempre della cartella *WorkFolder_ARS\ScambioDati* per poi tornare all'interno dello script principale.

Lato Matlab

In maniera del tutto sincrona, Matlab, che nel mentre si trova già in run nello script *ScambioDati_SA_ML.m*, cicla all'interno di un while (dal quale esce una volta verificata la condizione di uscita) in attesa del file *STARTMATLAB.txt* proveniente da SA. Una volta accertata la presenza del file di start all'interno della cartella di lavoro, si procede con l'eliminazione dello stesso e vengono lanciate le due function, *MMR.m* e *SMF.m*, che, come valori input, avranno tutti i dati contenuti nei file *XYZToMatlab.txt* e *ThetaOmegaToMatlab.txt*. Terminata l'esecuzione dei vari blocchi di codice si procede a:

- esportare le target position calcolate sul generico piano dal workspace di Matlab al file *PostionTarget.txt*;
- tornare all'interno del ciclo while contenuto nello script principale;
- generare il file *MatlabDone.txt* all'interno nella cartella *WorkFolder_ARS\ScambioDati*.

Lato SA

Mentre Matlab effettua il calcolo ed esporta i dati nel file di testo finale, SA si trova in un ciclo while di attesa con all'interno un delay di un secondo. Ad ogni secondo viene quindi fatta una verificata sull'esistenza del file *MatlabDone.txt* all'interno della cartella di lavoro. Nel momento in cui questa verifica da esito positivo, si procede all'eliminazione del file *MatlabDone.txt* per poi risalire il codice in modo da iniziare un nuovo ciclo di scansione relativo al piano successivo.

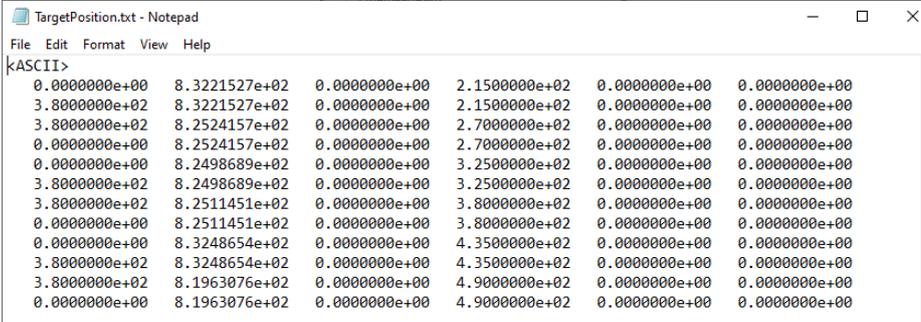
Terminato anche l'ultimo piano di scansione, lo script di SA viene indirizzato nel blocco di uscita nel quale viene creato un apposito file *SADone.txt* sempre nell'apposita directory e poi terminata l'esecuzione.

Lato Matlab

Terminato l'ultimo piano di ciclo non verrà più intercettato alcun file di start ma con la presenza del file *SADone.txt* viene verificata la condizione di uscita dal ciclo while e si ha l'arresto simultaneo dei due programmi.

2.4 File converter e creazione del file Avris

Terminata l'esecuzione di SpatialAnalyzer e Matlab occorre convertire il file *TargetPosition.txt*, che contiene le posizioni target da passare al software di controllo TwinCAT, in un file di tipo *TargetPosition.avr* leggibile da Avris. In figura 2.13, vi è un esempio di file generato da Matlab nel quale si possono scorgere sei differenti piani di scansione.



```
TargetPosition.txt - Notepad
File Edit Format View Help
kASCII>
0.000000e+00 8.3221527e+02 0.000000e+00 2.150000e+02 0.000000e+00 0.000000e+00
3.800000e+02 8.3221527e+02 0.000000e+00 2.150000e+02 0.000000e+00 0.000000e+00
3.800000e+02 8.2524157e+02 0.000000e+00 2.700000e+02 0.000000e+00 0.000000e+00
0.000000e+00 8.2524157e+02 0.000000e+00 2.700000e+02 0.000000e+00 0.000000e+00
0.000000e+00 8.2498689e+02 0.000000e+00 3.250000e+02 0.000000e+00 0.000000e+00
3.800000e+02 8.2498689e+02 0.000000e+00 3.250000e+02 0.000000e+00 0.000000e+00
3.800000e+02 8.2511451e+02 0.000000e+00 3.800000e+02 0.000000e+00 0.000000e+00
0.000000e+00 8.2511451e+02 0.000000e+00 3.800000e+02 0.000000e+00 0.000000e+00
0.000000e+00 8.3248654e+02 0.000000e+00 4.350000e+02 0.000000e+00 0.000000e+00
3.800000e+02 8.3248654e+02 0.000000e+00 4.350000e+02 0.000000e+00 0.000000e+00
3.800000e+02 8.1963076e+02 0.000000e+00 4.900000e+02 0.000000e+00 0.000000e+00
0.000000e+00 8.1963076e+02 0.000000e+00 4.900000e+02 0.000000e+00 0.000000e+00
```

Figura 2.13: File testo generato da Matlab con le target position da passare agli assi

Ogni riga del file ASCII contiene i sei valori target che a seconda del giunto di riferimento sono espressi in gradi o in millimetri. Questi valori saranno poi passati al controllo PLC per effettuare la movimentazione. A ciascun piano di scansione è associata una movimentazione, la quale viene sdoppiata all'interno di Matlab in due sotto movimentazioni:

- Con la prima riga, e le successive righe dispari, si procede alla movimentazione di tutti gli assi eccetto la tavola;
- Con la seconda riga, e le successive righe pari, si procede alla movimentazione della sola tavola rotante;

Questo approccio consente non solo una movimentazione più sicura e meno predisposta ad urti indesiderati ma permette di andare in scansione con il tool di misura solo quando tutti gli assi sono correttamente posizionati e con la tavola pronta a ruotare il pezzo. Ciò favorisce un tempo maggiore di assestamento della struttura e quindi una misurazione più pulita. In figura 2.14, viene invece mostrato il risultato dell'operazione di conversione del file testo in file Avris. Si può facilmente notare come a monte dei "Go To Joints" pari viene inserito uno start scan mentre a valle uno stop scan.

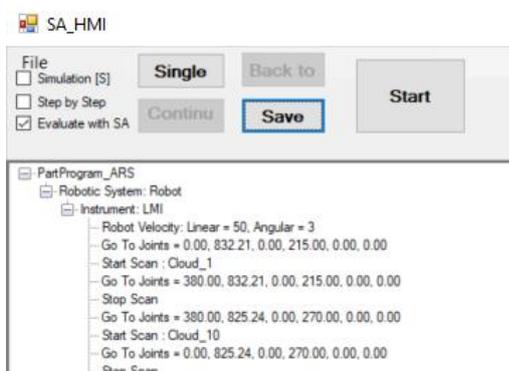


Figura 2.14: Esempio di file in formato Avris

2.5 Avris: albero ed interfaccia utente

Avris, che è il software sviluppato internamente da Axist, consente il simultaneo interfacciamento con il programma di collaudo dimensionale SpatialAnalyzer ed il controllo real-time PC-based di TwinCAT3. In figura 2.15 è visibile l'interfaccia utente di Avris. Cliccando sul tasto "Connect SA" in basso a sinistra, l'operatore si connette al file SA creato in precedenza per il calcolo delle position target. Per questo motivo, Avris, SA e Matlab devono coesistere sulla stessa macchina che prenderà il nome di PC-Server. Una volta connessi a SA, sarà quindi possibile importare nel software di collaudo le nuvole di punti acquisite tramite il tool di misura durante la scansione del componente. La corretta ricostruzione nello spazio delle varie nuvole di punti è assicurata sempre da Avris, che, dopo aver preso in carico i dati di scansione dal profilometro laser, applica una rototraslazione in funzione della cinematica internamente implementata e le posizioni attuali dei giunti provenienti dal controllo (TC3). Questo aspetto verrà meglio approfondito nel capitolo dedicato alla cinematica del sistema.

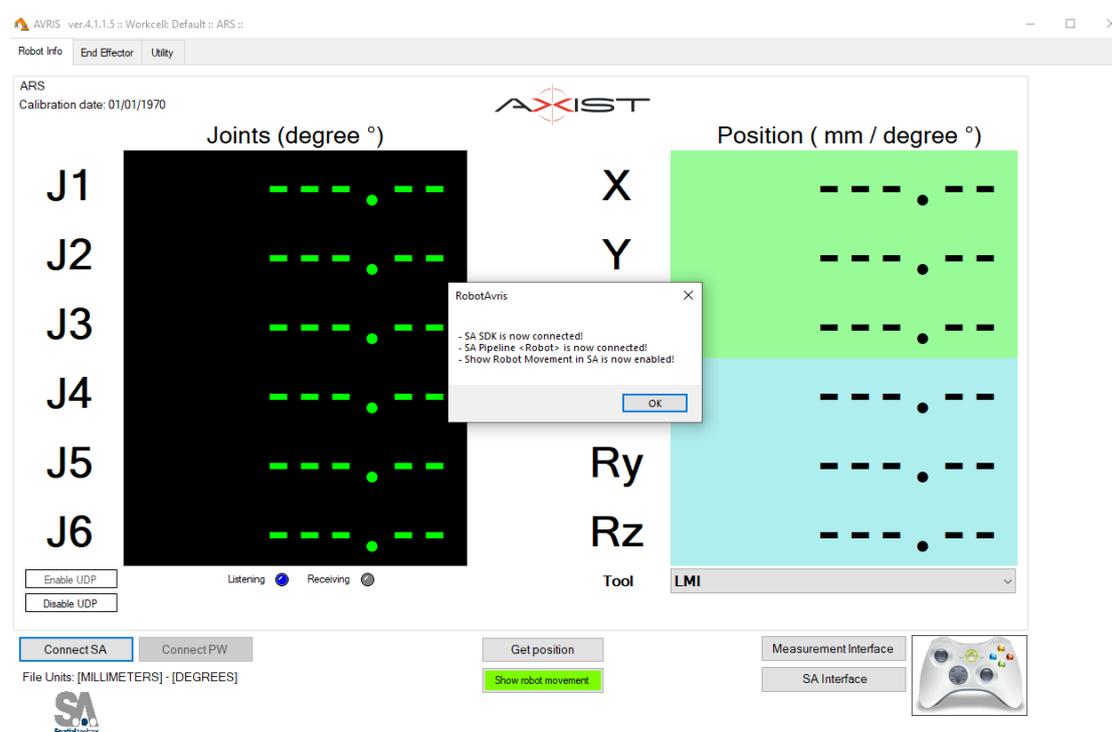


Figura 2.15: Interfaccia utente Avris

Per poter invece interfacciarsi con il software di controllo TwinCAT3 presente sul Pc-Beckhoff, a cui viene associato il nome di Pc-Client, si utilizza il file in formato avris di figura 2.14. Di seguito vengono spiegate le funzionalità della toolbar superiore di un file Avris base.

Simulation: se pinnato, permette di effettuare l'intera movimentazione solo in simulazione slegando di fatto il controllo dagli assi. La simulazione sarà visibile nel file SA e risulta molto utile laddove ci possono essere dubbi su interferenze ed urti a causa di pezzi con geometrie particolari;

Step by step: se pinnato, consente di scorrere l'albero in maniera graduale. Per poter andare avanti con il comando successivo occorre premere su *Continue*;

Evaluate with SA: permette di risincronizzare la posizione assoluta degli oggetti/frame con SA

Start: permette di avviare la movimentazione, che essa sia in simulazione o no;

Save: consente di salvare eventuali modifiche fatte sull'albero di scansione;

Facendo invece riferimento sempre alla figura 2.14, si elencano i comandi base che vengono "attivati" durante lo scorrimento dell'albero.

Robotic System: Robot, indica con quale sistema di misura presente in SA si è connessi. Il nome del sistema in SA e in Avris deve essere lo stesso;

Instrument: LMI, indica la tipologia di tool che è in uso;

Robot Velocity: Linear / Angular, consente di impostare la velocità di movimentazione degli assi lineari e rotazionali. I valori di default possono essere facilmente configurati dall'operatore tramite il comando *modify*;

Go to joints, comando che contiene i valori di posizione target da passare al controllo PC-based Beckhoff per la movimentazione assi;

Start Scan, attiva, lato SW, la modalità di scansione del profilometro laser;

Stop Scan, disattiva, lato SW, la modalità di scansione del profilometro laser.

A questi comandi base, che saranno quindi presenti in ogni file autogenerato, si affiancheranno una serie di funzionalità, sempre sviluppate dal settore ricerca e sviluppo, che consentono di:

- Creare dei pop up;
- Attivare subroutine;
- Effettuare operazioni di misura tra i valori nominali del CAD e quelli misurati;
- Generare file di reportistica;
- Esportare dati su file ASCII o file Excel;
- Importare dati da file esterni.

2.6 Configurazione della cartella "WorkFolder_ARS"

Come è possibile notare nei paragrafi precedenti, durante il procedimento di scansione vi sono delle azioni che per essere completate richiedono operazioni su determinati file. Per far ciò, è stato quindi necessario creare una cartella di lavoro che permettesse di avere delle directory fisse per i vari file usati e rendere quindi sicuro e ripetibile l'intero processo. In figura 2.16 viene quindi mostrata la "WorkFolder_ARS" in cui si distinguono le sottocartelle di:

- *Scambio dati*, all'interno della quale sono allocati i file per l'esportazione dei dati verso Matlab, i file necessari per il corretto funzionamento del protocollo SA-ML, il file di testo finale in cui vengono archiviate le target position calcolate;

- *ScriptARS*, all'interno della quale sono archiviati gli script in formato SA che vengono richiamati durante la fase iniziale di analisi del modello CAD. Per una facilità di debug e tracciabilità, è presente anche una sottocartella che contiene le vecchie versioni del codice.

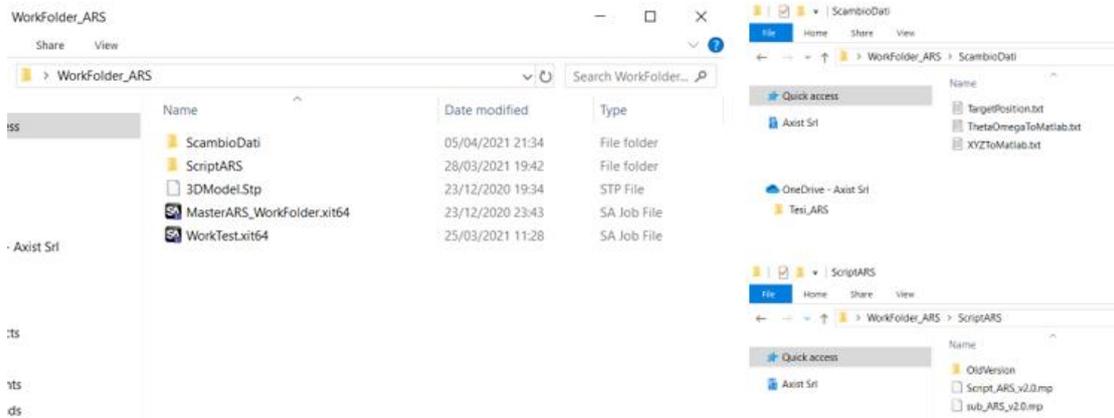


Figura 2.16: Workfolder_ARS

2.7 Controllo sistema in ambiente TwinCAT-Beckhoff

All'interno della solution "AvrisRotationalScanning.sln" vi è la gestione dei seguenti aspetti chiave del processo di scansione:

- Movimentazione degli assi;
- Gestione dell'evento trigger hardware verso gli encoder assi ed il tool di scansione;
- Scambio dati e variabili di controllo tra Pc-Server e Pc-Client;
- Gestione allarmi;
- Interfacciamento con HMI interna.

A questi aspetti se ne andranno ad aggiungere altri in una seconda fase di sviluppo:

- Interfacciamento con HMI esterna e customizzata sul sistema;
- Safety.

Nel seguente schema di figura 2.17 vengono rappresentati i canali di comunicazione implementati fra i vari sistemi in gioco e i pacchetti dati e le variabili che vengono scambiate durante l'esecuzione dei programmi.

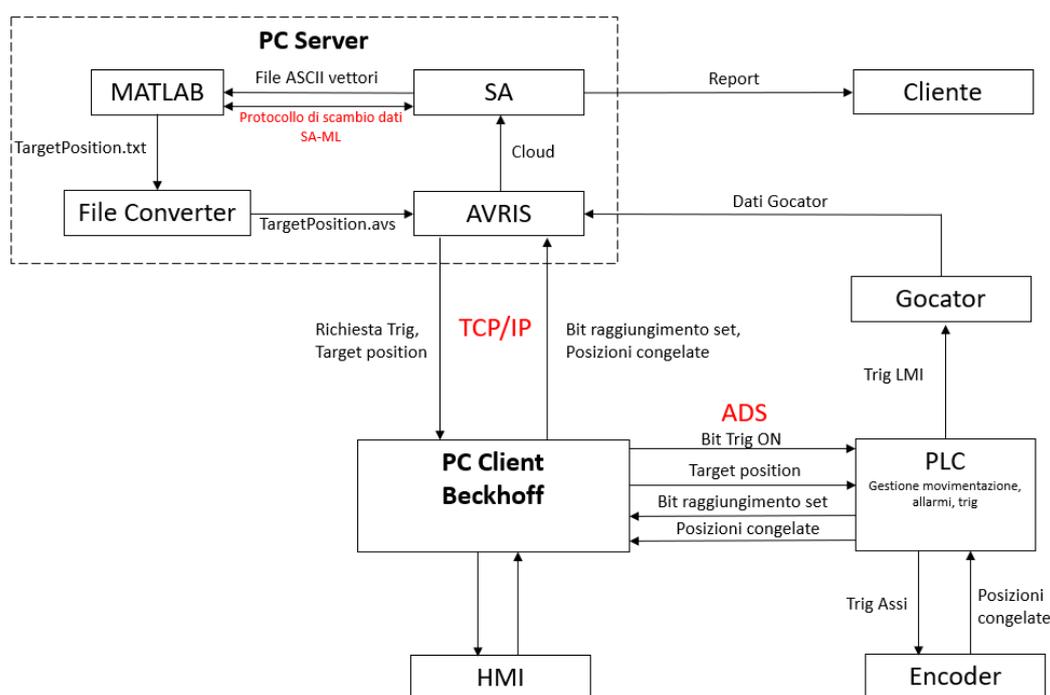


Figura 2.17: Interfacciamento PC-Server e PC-Client

All'interno del Pc-Server, durante la prima fase di generazione delle traiettorie, girano simultaneamente sia SpatialAnalyzer che Matlab. Durante invece la fase di scansione resta attivo solo il software di collaudo in modo da ricevere e visualizzare le nuvole di punti provenienti dal tool di scansione e ricostruite tramite la cinematica implementata in Avris. Sul Pc-Client Beckhoff, nel quale è presente il software di controllo TWINCAT3, si trova la Solution con il programma PLC e la comunicazione via ADS per l'invio e la ricezione dei dati.

Per entrare più nel dettaglio all'interno del programma PLC, occorre introdurre i vari blocchi funzionali che sono stati implementati per poter gestire le varie operazioni:

- fbAdsIO (FB), function block che si occupa della comunicazione Avris–TC3 e dell'evento Trigger;
- fbAxis (FB), function block della libreria Axist che si occupa del funzionamento del generico asse;
- fbCycleARS (FB), function block nella quale è implementato il ciclo macchina degli assi.

Tutte queste function block vengono richiamate all'interno del MAIN che costituisce la POU principale del programma. L'ordine con il quale vengono richiamate andrà a definire quella che sarà l'effettiva esecuzione del ciclo PLC ed il modo in cui verranno gestite le variabili di input e output delle FB e globali. Sono inoltre presenti, in ogni blocco, metodi e azioni per la gestione delle varie funzionalità interne e che verranno spiegati nei paragrafi successivi⁽¹⁾. ⁽¹⁾Per la descrizione dell'ambiente di sviluppo TwinCAT ci si rifà a quanto scritto nel capitolo 1.

2.7.1 Main

Contiene la task principale del ciclo PLC e vi sono due aree separate di programmazione:

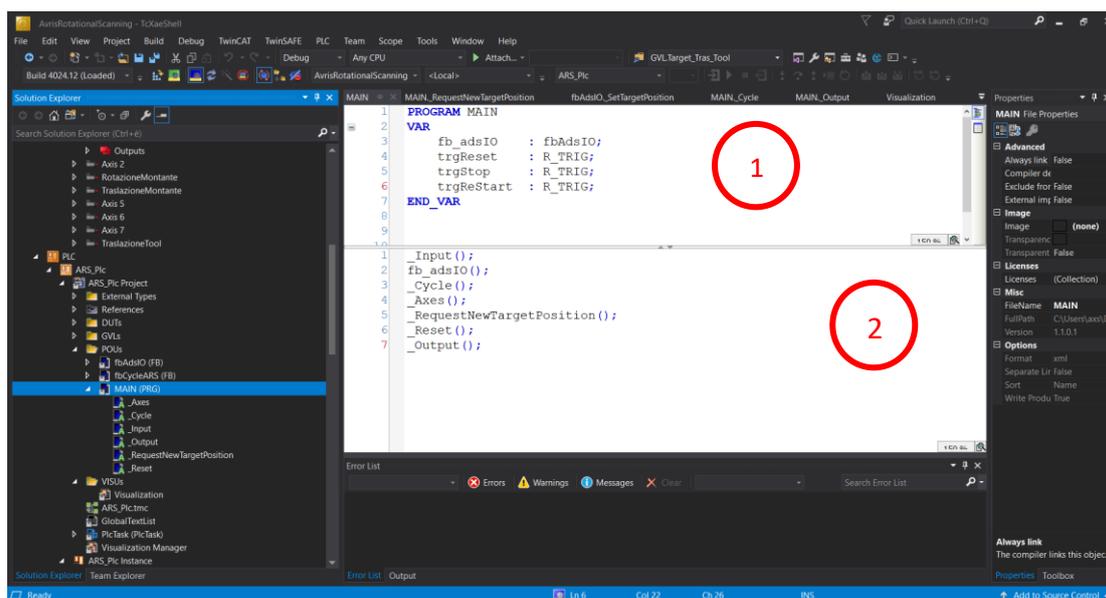


Figura 2.18: Main della solution

- 1- In questa area è possibile istanziare oggetti e dichiarare nuove variabili che visibili solo nel Main stesso. In questo caso applicativo, oltre alle variabili di tipo Trigger legate agli input provenienti dall'HMI, è importante sottolineare la creazione di un oggetto di tipo fbAdsIO che permetterà lo scambio dati con protocollo ADS, tra il PC-Client ed il PC-Server su cui girano Avris e SA;
- 2- Nell'area inferiore vengono richiamati tutti gli oggetti istanziati e le action necessarie a completare il ciclo PLC:

- L'Action _Input, che si occupa di aggiornare le variabili di ingresso delle altre action ad ogni ciclo di scansione;
- L'oggetto fb_AdsIO, oltre che della comunicazione ads, si occupa anche della creazione e gestione dell'evento trigger (sia lato software che lato hardware) e dell'aggiornamento delle posizioni target da passare agli assi;
- L'Action _Cycle, al cui interno sono istanziati gli oggetti di tipo fbCycleARS (un oggetto per asse). Durante l'esecuzione del ciclo plc vengono aggiornati gli input e gli output dei singoli oggetti;
- L'Action _Axes, al cui interno sono istanziati gli oggetti di tipo fbAxis (un oggetto per asse). Durante l'esecuzione del ciclo plc vengono aggiornati gli input e gli output dei singoli oggetti;
- L'Action _RequestNewTargetPosition che inoltra la richiesta di un nuovo set di posizioni andando ad alzare il bit ADS di avvenuta movimentazione (MoveDone).
- L'Action _Reset, che permette di resettare gli assi e tutte le variabili globali del ciclo in caso di allarmi ed errori macchina;
- L'Action _Output, che si occupa degli output in direzione dell'interfaccia PLC interna e del ciclo macchina.

2.7.2 fbAxis (FB)

Function Block presa direttamente dalla libreria AxiSt, creata appositamente per le commesse della settore AxiSt Robotics&Automation. All'interno della function sono già presenti una serie di funzionalità che consentono un controllo a trecentosessanta gradi dell'asse macchina. Ogni asse del sistema viene quindi istanziato (Declaration) come oggetto di tipo fbAxis all'interno delle variabili globali. In figura 2.19 viene mostrata come esempio la dichiarazione del primo asse relativo alla rotazione della tavola rotante.

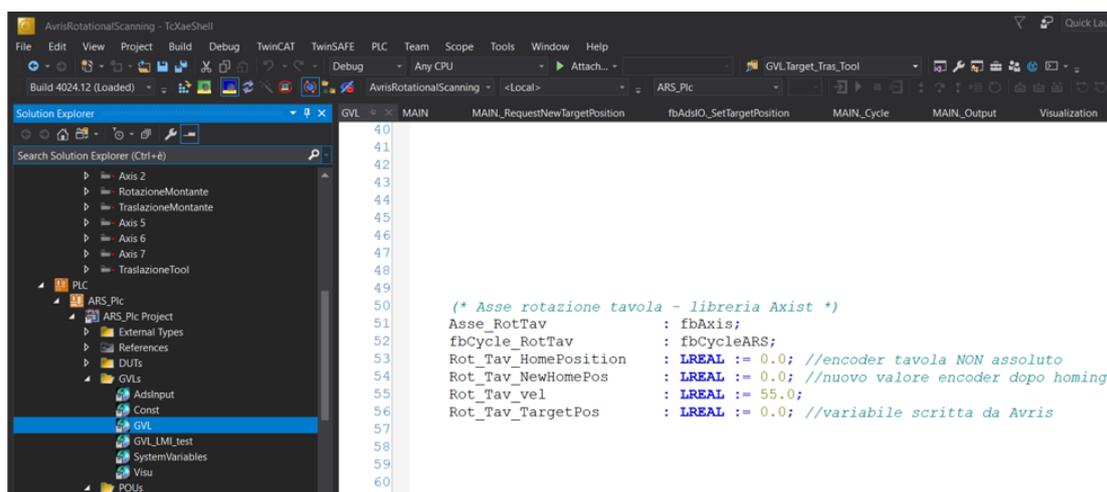


Figura 2.19: Dichiarazioni degli assi come oggetti di tipo fbAxis

La gestione dell'asse, sia fisico che virtuale, viene effettuata andando a richiamare (call) l'oggetto, che nel caso dell'esempio sarà "Asse_RotTav", durante l'esecuzione del tramite l'action _Axes del Main. All'interno dell'action per ogni ciclo PLC vengono aggiornate tutte le variabili di ingresso ai singoli assi se richiesto e lette quelle in uscita.

A seconda del tipo di variabili input che si andranno a scrivere, verranno attivate delle action definite internamente (l'ordine di attivazione segue quello di implementazione all'interno della fbAxis). Le action della fbAxis che vengono effettivamente utilizzate all'interno del progetto ARS sono:

- **_Input**, contiene i trig che attivano l'Homing, il Move, lo Stop ed il Reset;
- **_Move**, contiene la macchina a stati che gestisce la fase di move asse;
- **_SetNewPosition**, permette di settare la posizione attuale dell'asse con il valore desiderato;
- **_Jog**, contiene la macchina a stati che gestisce la fase di jog asse;
- **_Stop**, si occupa di stoppare la movimentazione in seguito ad una richiesta esterna di stop o in seguito ad un bit interno alzato;
- **_Reset**, si occupa del reset assi, che nel caso del sistema ARS coincide con il riportare gli assi nella rispettiva posizione di homing;
- **_Output**, permette di mandare in uscita alla function le variabili selezionate.

All'interno di ogni action è già presente una gestione interna degli allarmi e degli errori. Tramite l'assegnazione di variabili output è quindi possibile visualizzare, all'interno della finestra allarmi HMI, eventuali messaggi di allarme e/o malfunzionamenti.

2.7.3 fbCycleARS (FB)

Essendo la fbAxis paragonabili ad una black box, al cui interno sono già implementate tutte le funzionalità per il controllo di un generico asse, si è reso necessario andare a definire una seconda function che ne permettesse il corretto utilizzo e che consentisse, inoltre, di programmare l'effettivo ciclo di funzionamento degli assi tramite l'uso della macchina a stati⁽¹⁾. ⁽¹⁾*Definizione macchina a stati vedere capitolo 1, paragrafo 1.*

Per poter meglio spiegare quanto accade all'interno della function, risulta utile far riferimento allo schema a blocchi di figura 2.20. Con il colore nero vengono rappresentati i blocchetti facenti riferimento alla macchina a stati contenuta nella action “_AutoRun”. Tale action consente di eseguire il ciclo macchina, dalla fase di homing a quella di uscita, in maniera del tutto automatizzata ed integrata ad Avris. Tutti gli input trigger ed input “bistabili” provenienti dall'HMI sono invece rappresentati in blu. Le variabili verso e da fbAxis sono rappresentate con il colore verde:

- I bit in uscita relativi alle funzioni di Enable, Move, Stop, Reset, SetNewPosition;
- I bit in ingresso di fine esecuzione della movimentazione (MoveDone e MoveBusy), del set new position (SetDone), dello stop (StopDone);
- La variabile “TargetAsse” in uscita di tipo LREAL contenente la posizione da raggiungere.

In quest'ultima variabile, che viene passata al blocco che gestisce il Move del corrispettivo asse, a seconda dello stato macchina attivo è caricato il valore (write) di una delle seguenti variabili:

- TargetHome, posizione di sicurezza (homing) che ciascun asse deve raggiungere non appena viene attivata la scansione;
- TargetAvris, posizione che viene passata da Avris e che si aggiorna con il progredire della scansione;

- EndCycletarget, posizione che consente la movimentazione di uscita in tutta sicurezza dglì assi una volta terminata la scansione.



Figura 2.20: macchina a stati, action `_AutoRun`

Continuando a seguire lo schema colore, con il colore rosso sono rappresentati i bit in ingresso relativi alla gestione delle posizioni target provenienti da Avris mentre in arancione sono indicati le variabili booleane, in ingresso ed in uscita, che permettono l'esecuzione della movimentazione finale e fermo macchina.

Come si può notare, la `fbCycle`, oltre ad avere al suo interno la macchina a stati che gestisce l'asse, fa da "ponte" fra le varie funzioni presenti all'interno del codice PLC e permette di avere una manutenibilità più semplice del codice stesso.

2.7.4 Gestione delle posizioni target

Durante la fase di scansione, ovvero dopo aver effettuato la fase di homing, per attivare la movimentazione degli assi occorre passare di volta in volta la posizione target come variabile di ingresso all'oggetto di tipo fbAxis. Il valore della posizione da raggiungere viene inviato da Avris (PC-Server) a TwinCAT (PC-Client) tramite comunicazione ADS. Per fare ciò si è reso necessario creare una variabile di tipo ARRAY, avente lo scopo di archiviare i valori di tipo provenienti da Avris. Nella function block fbAdsIO si trova l'action “_SetTargetPosition” che, richiamando il corrispettivo metodo, si occupa della scrittura dei valori target sulle variabili PLC che sono quelle che andranno ad interfacciarsi con i blocchi di movimento, figura 2.21.



Figura 2.21: processo di sovrascrittura delle variabili di target PLC

Per poter importare i valori delle posizioni target da un software esterno si sfrutta l'architettura TwinCAT che consente di trattare i singoli moduli del software (es: PLC, NC, ecc.) come dispositivi indipendenti. I messaggi tra questi oggetti e con le connessioni TCP / IP esterne vengono scambiati tramite un'interfaccia ADS. L'Automation Device Specification è un'interfaccia indipendente dal dispositivo e dal bus di campo capace di regolare il tipo di accesso ai dispositivi ADS. A questo punto, per eseguire la comunicazione ADS in maniera corretta, si è reso necessario creare le seguenti variabili apposite:

ADS input

Variabili inviate da Avris verso il progetto PLC. Esse sono costituite da:

- L'array di archiviazione target position;
- Il bit di avvenuto caricamento delle target position;
- Il bit di fine esecuzione Avris.

ADS output

Variabili inviate dal PLC verso Avris. Esse sono costituite da:

- Bit di fine movimentazione asse;
- Bit di errore.

I bit di entrata uscita consentono di definire quando è possibile caricare via ADS una nuova posizione target e quando è possibile sovrascriverla sulle variabili PLC. In particolare, si ha il seguente flusso logico:

1. Avvio di Avris e Start macchina. Non è necessario l'avvio simultaneo o un ordine di start prestabilito.
2. Durante la fase di start, viene eseguito dal PLC il movimento di homing mentre Avris inizia a scorrere lungo l'albero fino ad intercettare il primo go to joints.

3. Avris carica le prime posizione target all'interno dell'apposito Array ed alza il bit di avvenuto caricamento.
4. Lato PLC possono succedere due eventi:
 - La movimentazione di homing è terminata prima del caricamento di Avris;
 - La movimentazione di homing è ancora in esecuzione.

Nel primo caso è possibile accedere all'action che si occupa della scrittura dei valori sulle variabili PLC mentre nel secondo caso si resta in attesa che la fase di homing sia ultimata e che venga alzato il Bit di Homing done;

5. Caricate le posizioni sulle variabili PLC, inizia la movimentazione assi. Durante questa fase Avris è in attesa e non scorre lungo l'albero;
6. Terminata la fase di movimentazione il PLC alza il bit Ads di movimentazione terminata. Tale Bit viene letto da Avris che quindi avanza lungo l'albero fino a che non intercetta il go to joint successivo;

Nel momento in cui AVRIS arriva al comando di fine movimentazione "Move RotoSystem to: REAR" viene alzato il Bit Ads di Avris Done. Il PLC, letto questo Bit, fa partire la fase di uscita della macchina a stati ed inizia la movimentazione finale che con l'allontanamento della tavola dal tool di misura permette di mettere in sicurezza il componente ed il sistema.

2.7.5 fbAdsIO

All'interno di questa function sono presenti le azioni ed i metodi che permettono di:

- Settare le target position provenienti da AVRIS tramite ADS sulle variabili PLC;
- Gestire l'attivazione del trigger hardware;
- Raccogliere le posizioni "congelate" dei giunti in seguito al trigger di scansione e le actual position da passare a SA per la simulazione online.

La gestione del trigger hardware verso il profilometro laser e verso i drive dei motori avviene tramite il modulo Beckhoff EL2809 di digital output a 16 uscite. A seguito del comando di "start scan" AVRIS alza tramite ADS il bit di "execute trig" che viene letto dal PLC. Successivamente vengono mandati i segnali trigger e lato PLC si attende l'avvenuto congelamento delle posizioni mentre lato Gocator si attende l'importazione dei punti relativi alla lama di scansione. Nel momento in cui AVRIS riceve entrambi i pacchetti esporta la lama di scansione, ricostruita con la cinematica corretta, su SA e manda una nuova richiesta di esecuzione trigger andando a chiudere il cerchio.

2.8 Interfaccia HMI interna al progetto PLC

Durante la fase di sviluppo del codice PLC si è reso più volte indispensabile effettuare test mirati sia sulla movimentazione che sulle varie FB interne. Per facilitare quindi l'attività di testing è stata sfruttata la possibilità fornita dall'ambiente di sviluppo TwinCAT di costruire una interfaccia interna al progetto PLC. Questa interfaccia, seppur costituita dai soli elementi essenziali, ha permesso di interfacciarsi con il codice in maniera diretta e veloce senza dover forzare manualmente il cambio di variabili di stato o cercare la variabile desiderata. In figura 2.22 viene mostrata l'interfaccia prima di mandare in run il ciclo macchina e attivare la movimentazione.

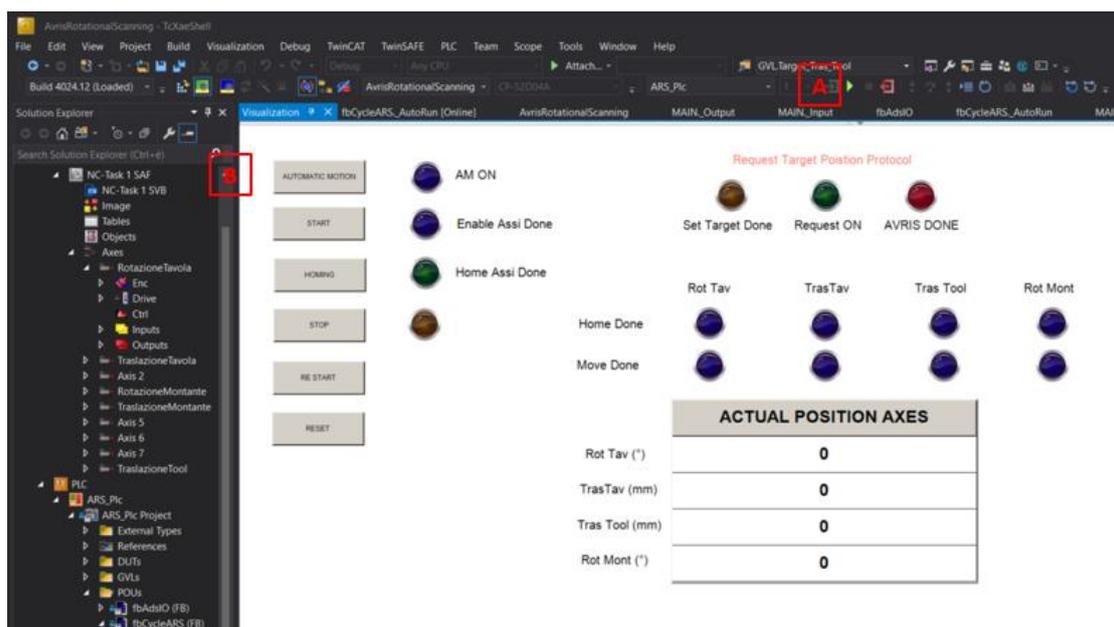


Figura 2.22: HMI nella configurazioni iniziale

Una volta dato lo start al PLC tramite il comando (A), per poter effettuare l'attivazione degli assi è necessario premere il pulsante di "Automatic Motion" (B) per entrare nella modalità di movimentazione guidata da AVRIS ed il comando di START. La spia "Enable Assi Done" si attiva quando tutti gli assi sono correttamente abilitati e pronti a muoversi. Nella tabella "Actual Position Axes" vengono aggiornati i valori delle posizioni degli assi ad ogni ciclo di scansione e permettono all'operatore di verificarne la posizione anche da remoto o comunque senza restare nei pressi della macchina. A questo punto dando il comando di "Homing" viene comandato agli assi di raggiungere la propria posizione safe in quanto potrebbero essere stati fermati in una posizione generica a causa di uno stop di emergenza. L'avvenuto completamento della fase di homing viene segnalato dall'accensione del led "Home Assi Done" (C) mentre l'homing relativo al singolo asse è visibile tramite i led. Alla fine di ogni movimentazione, compresa quella per effettuare l'homing, vengono attivati anche i led di Move Done (D).

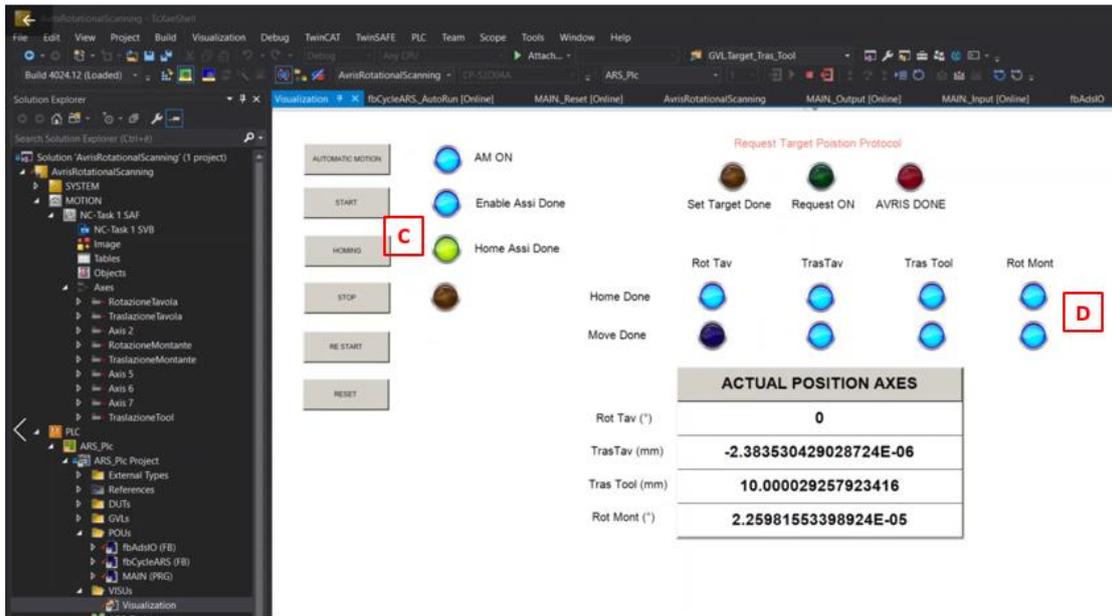


Figura 2.23: HMI ad homing effettuato

Effettuato l’homing, il sistema resta in attesa delle target position da AVRIS e del bit di inizio movimento. In alto a destra, sono presenti i led relativi al protocollo di comunicazione ADS e mostrano in tempo reale lo stato delle variabili dal PC-Client Beckhoff al PC-Server con AVRIS e viceversa. Il bit di fine esecuzione proveniente da AVRIS viene visualizzato dal led “AVRIS DONE” (E) e resta alto fino a quando non viene terminata la movimentazione di uscita e disimpegno pezzo, figura 2.24.

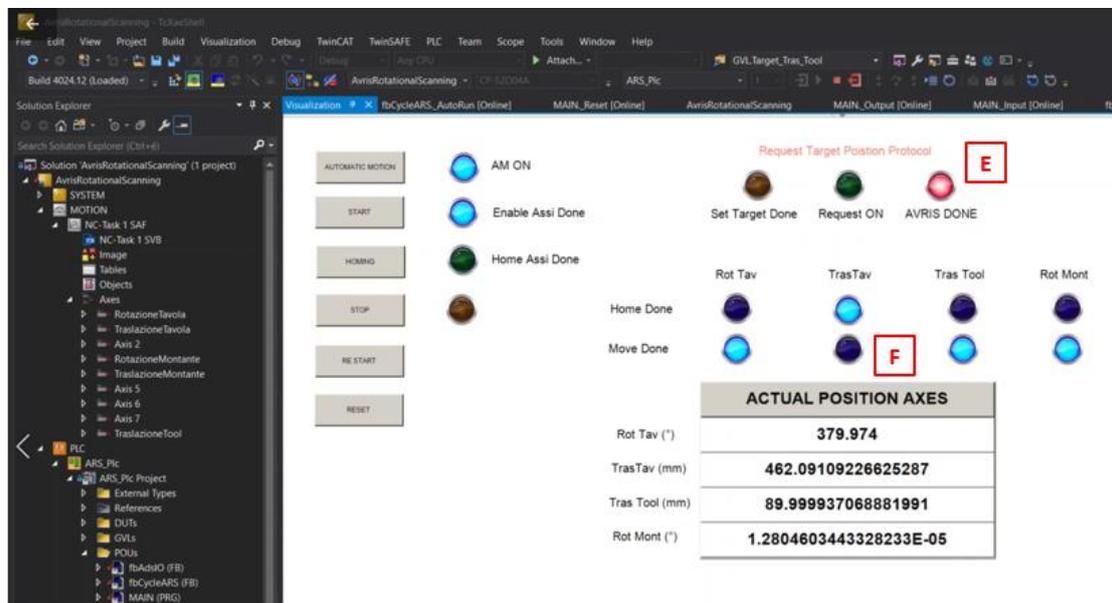


Figura 2.24: HMI movimentazione di fine ciclo

Come si può notare, durante la fase di uscita viene allontanata la tavola rotante dal profilometro laser senza movimentare il tool e di conseguenza l’unico led di “Move Done” non ancora alto è proprio quello relativo alla traslazione della tavola (F). Terminata la movimentazione di uscita, per far ripartire nuovamente il sistema occorre effettuare nuovamente la sequenza di start e homing per evitare movimentazioni

indesiderate e aumentare la sicurezza in fase di testing. Oltre ai comandi e led analizzati ve ne sono altri ausiliari che consentono di fermare e far ripartire il sistema in caso di bisogno (STOP e RESTART), ai quali è associato anche un led per monitorare l'attivazione, un comando di RESET che provvede a resettare non solo le variabili di ciclo macchina ma anche gli azionamenti in caso di errore nel PLC o nel drive Kollmorgen.

3. Modello cinematico ARS

Come descritto nei capitoli precedenti, la corretta localizzazione nello spazio delle singole “lame” di scansione viene effettuata tramite il simultaneo interfacciamento del software proprietario Avris con:

- il software di collaudo dimensionale SpatialAnalyzer;
- i dati degli encoder provenienti dal PC-Client Beckhoff;
- i dati provenienti dal tool di scansione.

All'interno della workcell di Avris, la quale verrà meglio descritta nei paragrafi successivi, è infatti possibile gestire la cinematica del sistema di misurazione tramite l'inserimento di parametri cinematici calcolati secondo la convezione di Denavit Hartenberg standard. Ad Avris si affianca in parallelo SpatialAnalyzer che, oltre alle attività di misurazione e creazione del report di collaudo finale, permette di gestire al proprio interno un modello CAD dello strumento di misura (in questo caso sistema ARS e tool di misura) e la cinematica ad esso associata. Ciò permette di mandare in simulazione la movimentazione del sistema sia online che offline e di fornire all'operatore un ambiente di sviluppo maggiormente intuitivo.

Prima di entrare nello studio della cinematica del sistema ARS è necessario analizzare due aspetti, strettamente interconnessi tra loro, che hanno indirizzato lo sviluppo e l'implementazione del modello cinematico sui vari programmi. Il primo aspetto, di natura software, nasce proprio dall'esigenza/obiettivo di integrare il sistema ARS all'interno di AVRIS. Tale programma, lanciato sul mercato da ormai diversi anni, nasce per la gestione di robot antropomorfi a sei gradi di libertà. Nonostante un continuo lavoro e sviluppo dell'area R&D, la capacità di adattamento del codice sorgente rispetto a sistemi di natura diversa da quelli antropomorfi risultava essere, al momento dello sviluppo iniziale, limitata in alcuni punti chiave:

- Gestione di una sola macchina per volta;
- Inserimento di un'unica catena cinematica;
- Numero di gradi di libertà da gestire pari a sei.

Queste caratteristiche di sistema, ininfluenti nella gestione dei bracci robotici, diventano fortemente limitanti se associate alla struttura del sistema fisico.

L'ARS è infatti scomponibile in due sottosistemi indipendenti:

- Sottosistema tavola;
- Sottosistema montante:

Il primo sottosistema è caratterizzato da due gradi di libertà corrispondenti alla rototraslazione della tavola sulla quale viene fissato il componente da misurare. Il secondo sottosistema presenta invece tre gradi di libertà, aventi lo scopo di orientare nello spazio il tool di misura. Risulta quindi evidente la presenza di due sistemi ben distinti, ciascuno caratterizzato da una propria cinematica indipendente. In figura 3.1 viene riproposto lo schema costruttivo preso dalla manualistica ufficiale Gudel, vedere paragrafo 1.3, dal quale sono state eliminate le movimentazioni non usate (traslazione montante e traslazione del secondo carrello verticale lungo la verticale) e aggiunta la movimentazione riguardante la rotazione della tavola (posizionata sul carrello

orizzontale). Si fa notare, inoltre, che nella struttura originale non è presente l'asse per la gestione della rotazione del tool. Tale asse è stato comunque considerato durante lo studio della cinematica del sistema in quanto, per poter effettuare dei test, ne è stata implementata una configurazione temporanea e priva di azionamento.

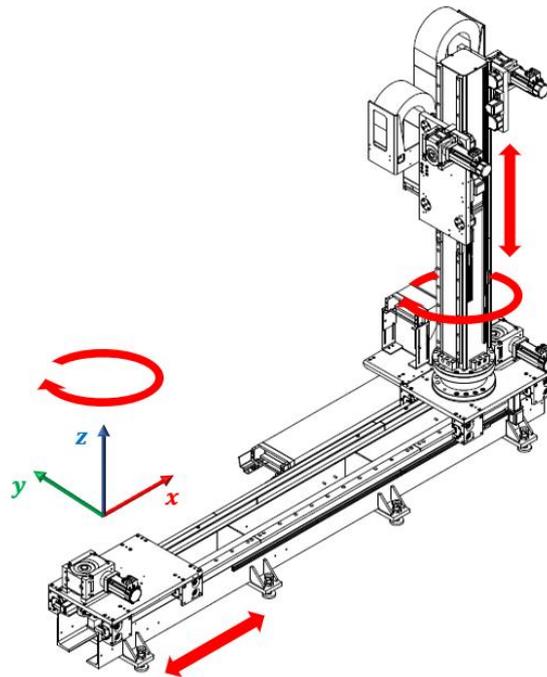


Figura 3.1: schema guide Gudel con movimentazioni effettivamente usate

Per risolvere quindi le due problematiche presentate in precedenza, è stato necessario implementare un nuovo modello caratterizzato da un'unica catena cinematica. Tale modello, che differisce quindi dal sistema fisico reale ma ne è assolutamente equivalente da un punto di vista cinematico, è stato sviluppato partendo dalle seguenti assunzioni: invece di lasciare il grado di libertà della rotazione alla tavola girevole, si considera il moto relativo del montante rispetto alla tavola stessa. Quest'ultima nel modello sarà quindi considerata fissa, ovvero, invece di movimentare direttamente l'oggetto da misurare sulla tavola, si fa ruotare il montante ed il tool di misura rispetto al centro tavola. Un approccio analogo è stato applicato alla traslazione del montante rispetto alla tavola: se nella movimentazione reale sarà la tavola a traslare verso il montante, che rimane fisso, nel modello cinematico si mantiene la tavola ferma e si avvicina il montante. Nella *tabella 3.1* sono riassunte le differenze tra il sistema reale ed il modello cinematico sviluppato.

Avris Rotational Scanning	ARS Kinematics Model
<i>Due sottosistemi indipendenti (2+3 gdl)</i>	<i>Unico sistema (5 gdl)</i>
<i>Doppia catena cinematica</i>	<i>Cinematica univoca</i>
<i>Rotazione sottogruppo tavola</i>	<i>Rotazione montante e tavola fissa</i>
<i>Traslazione tavola</i>	<i>Traslazione montante e tavola fissa</i>

Tabella 1 - Differenze tra sistema fisico e modello cinematico

In figura 3.2 sono schematizzati i giunti ed i link del modello cinematico secondo la rappresentazione convenzionale. Si nota come nel modello cinematico siano presenti due link “fittizi” che collegano i primi giunti in modo da ottenere, come detto in precedenza, una catena cinematica unica che colleghi tutti i gradi di libertà in gioco.

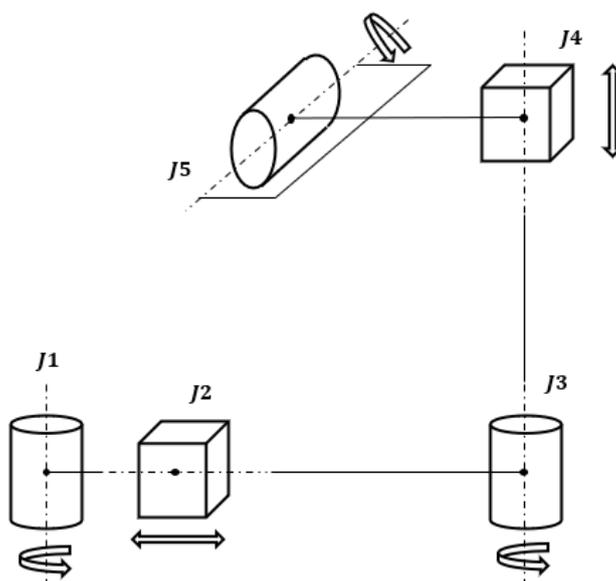


Figura 3.2 -Schema del modello cinematico

Una volta indentificato il modello base è ora possibile andarne ad effettuare lo studio cinematico. Per far ciò, è necessario definire la posizione e l'orientazione relativa dei singoli link tramite la determinazione delle frames in corrispondenza di ogni giunto della struttura. Per la definizione della frame i -esima è stata seguita la convenzione standard di *Denavit Hartenberg*, la quale fornisce le seguenti regole:

- Scegliere l'asse z_i lungo l'asse del giunto $i+1$;
- Posizionare l'origine O_i all'intersezione dell'asse z_i con la normale comune agli assi z_i e z_{i-1} ;
- Scegliere l'asse x_i sempre lungo la normale comune e con direzione che va dal giunto i al giunto $i+1$;
- Scegliere l'asse y_i in modo tale da formare una terna destrorsa.

La convenzione presenta alcune eccezioni per casi particolari e che vengono riportate in quanto presenti nel modello usato:

- Quando due giunti consecutivi presentano assi paralleli, la normale comune non è univocamente definita;
- Quando due assi consecutivi si intersecano, la direzione di x_i è arbitraria;
- Quanto il giunto i è prismatico, la direzione dell'asse di z_{i-1} è arbitraria.

Note le regole, si elencano i cinque assi nonché gradi di libertà del modello cinematico:

- J1 – asse rotoidale
- J2 – asse prismatico
- J3 – asse rotoidale
- J4 – asse prismatico
- J5 – asse rotoidale

Tenendo presente che la numerazione delle frames è scalata di uno rispetto alla numerazione dei giunti, rispettando le regole definite dalla convenzione DH e ponendo la terna base in corrispondenza del primo giunto, sono state ottenute le terne rappresentate in figura 3.3.

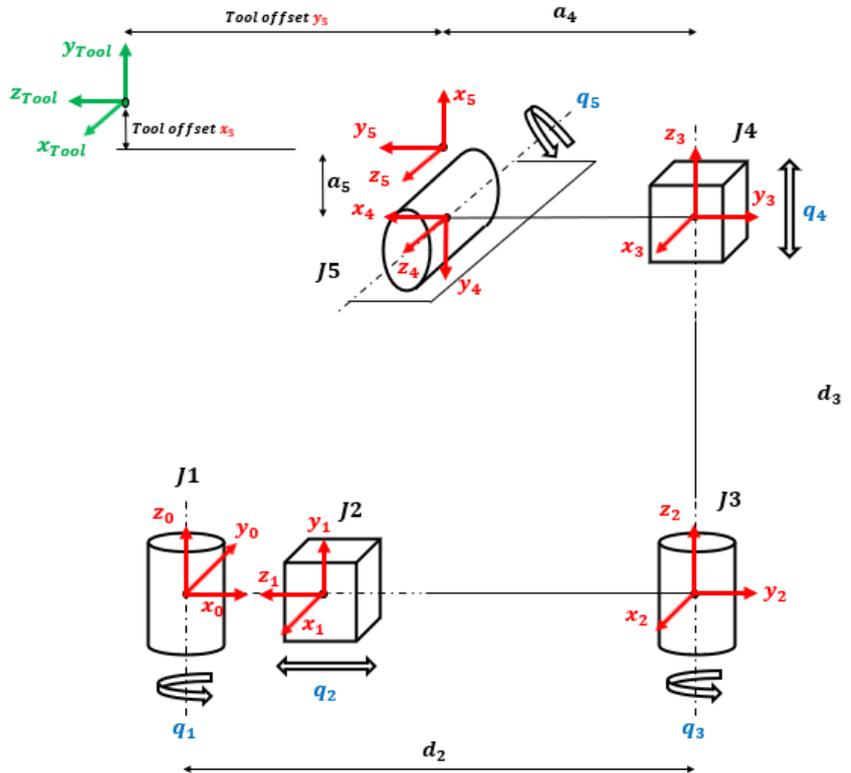


Figura.3.3- Frames secondo la convenzione DH

Il sistema di riferimento inerziale del modello (x_b, y_b, z_b) , detto di base, non è rappresentato in figura poiché centrato anch'esso nell'origine della prima terna di riferimento (x_0, y_0, z_0) e con gli assi sovrapposti.

Una volta disposte le terne, risulta essere più intuitiva identificare la semplificazione introdotta dal modello cinematico:

- il primo giunto rotoidale J1 imprime una rotazione a tutto ciò che si trova a valle, di conseguenza sarà il “sottogruppo” montante costituito dagli ultimi quattro giunti a ruotare rispetto all’asse Z del sistema di riferimento base;
- il giunto prismatico J2 presenta origine comune a J1 ed infatti non ha alcun parametro di offset rispetto la prima terna. Per comodità di rappresentazione vengono rappresentati distanziati mentre il parametro d_2 corrisponde alla distanza massima tra l’asse della tavola e l’asse del montante quando la prima è in posizione di zero (origine O_0 coincidente con O_b). Il controllo invierà ad Avris la posizione attuale della tavola rotante ad ogni scansione del profilometro laser in modo da ricostruire la reale distanza relativa tra montante e tavola;
- I giunti J3, J4 e J5 non differiscono rispetto alla configurazione reale e rappresentano infatti la rotazione effettiva del montante, la traslazione del tool tramite il carrello verticale e la rotazione del tool di misura.

Per determinare la posizione e l'orientazione della frame $i-1$ rispetto alla frame i , si utilizzano i quattro parametri di *Denavit Hartenberg*:

- a_i , distanza di O_i rispetto a O_{i-1} lungo l'asse x_i ;
- d_i , coordinata di O_i rispetto a z_{i-1} ;
- α_i , rotazione per portare l'asse z_{i-1} sull'asse z_i rispetto a x_i , positiva se antioraria;
- ϑ_i , rotazione per portare l'asse x_{i-1} sull'asse x_i rispetto a z_{i-1} , positiva se antioraria;

I parametri a_i e α_i sono sempre costanti mentre, a seconda del tipo di giunto, uno dei parametri tra d_i e ϑ_i sarà la variabile di giunto. In particolare, si ha che:

- se il giunto è rotoideale, la variabile è su ϑ_i
- se il giunto è prismatico, la variabile è su a_i

È possibile definire una tabella, dove ad ogni link (e giunto) corrisponde una riga all'interno della quale sono presenti i parametri caratteristici del link e del giunto i -esimo. La tabella 3.2 è fondamentale per il calcolo della cinematica diretta del robot, ovvero della funzione che permette di trasformare le coordinate dal sistema dell'end effector (tool di misurazione) al sistema di coordinate della base.

	a_i	d_i	α_i	ϑ_i
0-1	0	0	$\pi/2$	$-\pi/2 + q_1$
1-2	0	$d_2 + q_2$	$-\pi/2$	0
2-3	0	d_3	0	q_3
3-4	a_4	q_4	$-\pi/2$	$-\pi/2$
4-5	a_5	0	0	$-\pi/2 + q_5$

Tabella 2 - Parametri di Denavit Hartenberg del modello cinematico

La terna cinque di figura 3.3 fa riferimento all'orientazione e posizione della flangia sulla quale viene poi fissato il tool di misura. L'ultima frame, rappresentata con il colore verde, costituisce il sistema di riferimento del centro del trapezio di scansione per AVRIS. Tale sistema, infatti, non coincide con quello reale del profilometro laser fornito dalla casa produttrice (Gocator LMI 2430) ma segue la seguente convenzione interna:

- Asse Z in direzione del fascio laser;
- Asse X in direzione opposta ai connettori;
- Asse Y disposto in modo da chiudere la terna destrorsa.

All'interno di Avrìs è implementata una matrice di riflessione che permette di rototraslare i punti dal sistema di riferimento "centro trapezio LMI" al sistema di riferimento "centro trapezio Avrìs", figura 3.4. Tale scelta nasce dal fatto che all'interno di AVRIS è possibile gestire diverse tipologie di tool di misura, ciascuno dei quali si andrà ad interfacciare con una frame normalizzata e generica.

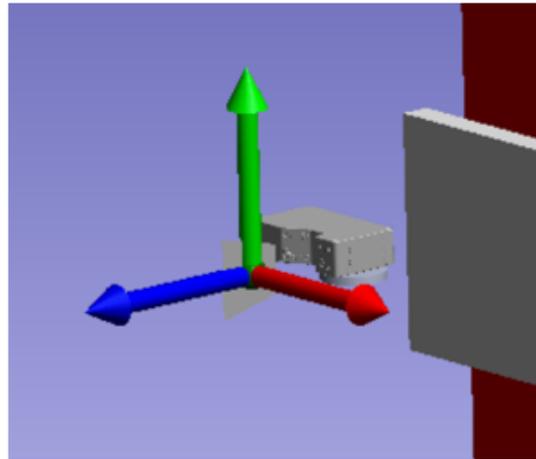
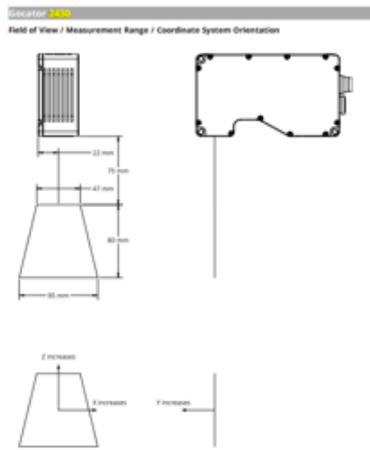


Figura 3.4: A sinistra terna centro trapezio LMI, a destra terna centro trapezio Avris

Per la validazione del modello cinematico ottenuto e dei parametri di DH ad esso associati, è stato sviluppato uno script in Matlab, “CinematicaARS.m”. All’interno dello script sono presenti le varie matrici omogenee di rototraslazione seguendo la stessa convenzione presente in Avris. Inserendo nello script pochi dati input, quali le variabili di giunto e gli offset fissi, si è verificato il corretto funzionamento del modello cinematico, figura 3.5.

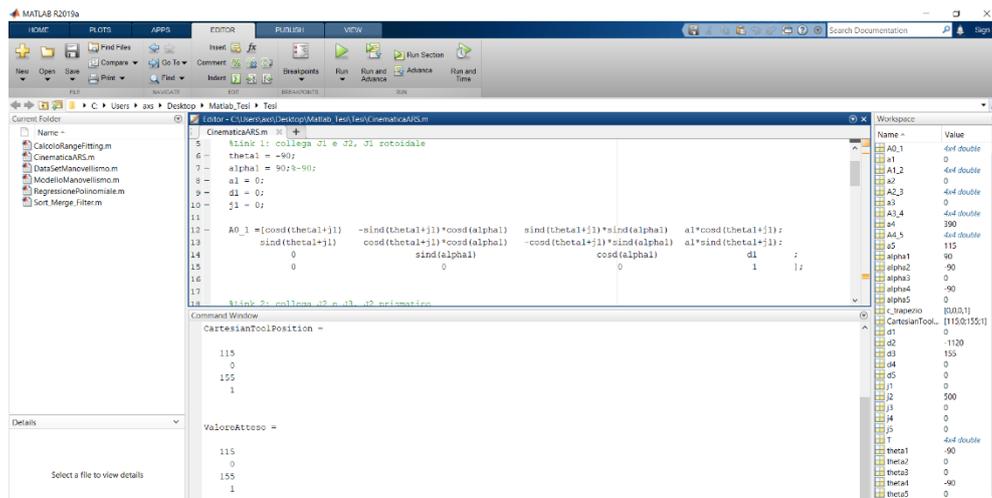


Figura.3.5: Script di verifica del modello cinematico

3.1 Implementazione della cinematica in AVRIS

Per poter sfruttare la cinematica calcolata nel paragrafo precedente ed inserire i vari parametri trovati all'interno di AVRIS, è necessario andare a definire un Robotic System che rappresenta il "corpo" principale del sistema. Il Robotic System è composto da tre elementi:

- Manipolatore;
- End Effector;
- External Tracking System (opzionale).

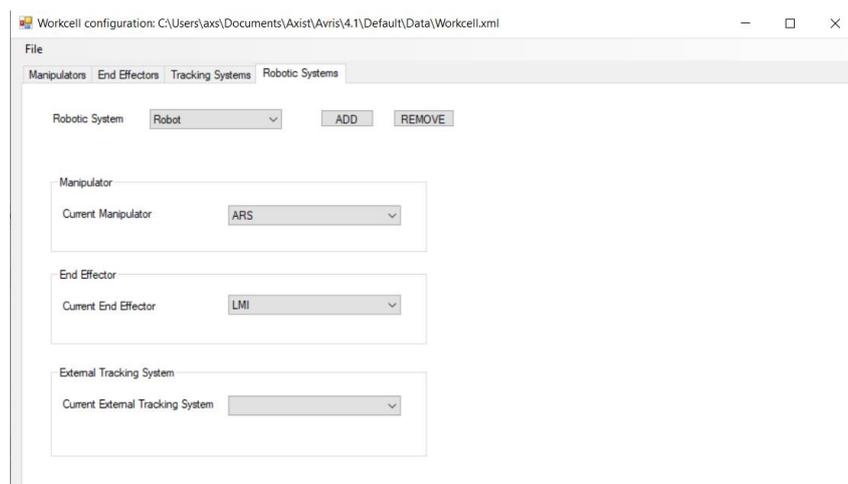


Figura 3.6 - Robotic System, workcell di Avris

Come sarà più chiaro in seguito, la configurazione della workcell permette di:

- applicare ai punti misurati la corretta catena cinematica e quindi di importarli correttamente nel sistema di riferimento base del file di SpatialAnalyzer. Ciò permette di effettuare le attività di misurazione e collaudo dimensionale tra valori nominali e misurati;
- inviare al file SA i parametri cinematici calibrati per rendere la movimentazione simulata (online o offline) quanto più vicina possibile alla movimentazione reale e facilitare di conseguenza la fase di testing prima della messa in funzione della macchina.

3.1.1 Manipolatore

Robot e/o sistema del quale si vuole controllare la movimentazione e quindi gestire la cinematica. Nel caso di Robot antropomorfi sono già presenti in AVRIS i manipolatori che vengono generalmente usati all'interno della divisione Axist Robotics e di cui sono già settati i parametri di DH nominali. Nel caso del sistema ARS si è quindi dovuto procedere con la creazione di un manipolatore apposito costituito da tre giunti rotoidali (J1, J3, J5), due giunti prismatici (J2, J4) ed un sesto giunto fantasma. Tale ultimo giunto si è reso necessario in quanto AVRIS è sviluppato per la gestione di un manipolatore con almeno sei gradi di libertà.

Per ogni giunto sono stati inseriti i parametri già riportati nella tabella 3.2. Viene proposta in figura 3.7 l'interfaccia per l'inserimento dei parametri nominali del primo giunto rotoidale, per tutti gli altri giunti si procede in modo analogo.

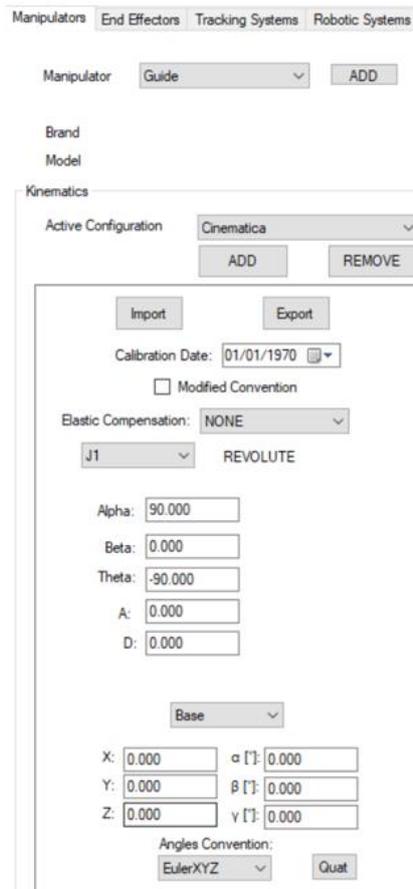


Figura 3.7 - Kinematics interface, workcell di Avrìs

La matrice di trasformazione implementata in AVRIS per i giunti rotoidali è del tipo:

$$\hat{A}_1^{i-1} \begin{bmatrix} \cos(\vartheta_i + q_i) & -\sin(\vartheta_i + q_i) \cos \alpha_i & \sin(\vartheta_i + q_i) \sin \alpha_i & a \cos(\vartheta_i + q_i) \\ \sin(\vartheta_i + q_i) & \cos(\vartheta_i + q_i) \cos \alpha_i & -\cos(\vartheta_i + q_i) \sin \alpha_i & a \sin(\vartheta_i + q_i) \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Mentre per i giunti prismatici:

$$\hat{A}_1^{i-1} = \begin{bmatrix} \cos \vartheta_i & -\sin \vartheta_i \cos \alpha_i & \sin \vartheta_i \sin \alpha_i & a \cos \vartheta_i \\ \sin \vartheta_i & \cos \vartheta_i \cos \alpha_i & -\cos \vartheta_i \sin \alpha_i & a \sin \vartheta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i + q_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Si ricorda che le precedenti matrici di rototraslazione sono il risultato della post moltiplicazione delle seguenti trasformazioni:

- Traslazione della terna i-1 di una lunghezza pari a d_i lungo l'asse z_{i-1} e rotazione di ϑ_i rispetto l'asse z_{i-1} . Questa sequenza permette di passare dal frame i-1 ad un frame i' ed è descritta dalla seguente matrice di trasformazione omogenea

$$\hat{A}_{i'}^{i-1} = \begin{bmatrix} \cos \vartheta_i & -\sin \vartheta_i & 0 & 0 \\ \sin \vartheta_i & \cos \vartheta_i & 0 & 0 \\ 0 & 0 & 0 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Traslazione della terna i' di una lunghezza pari a a_i lungo l'asse x_i e rotazione di α_i rispetto l'asse x_i . Questa sequenza permette di passare dal frame i' al frame i ed è descritta dalla seguente matrice di trasformazione omogenea

$$\hat{A}_i^{i'} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos \vartheta_i & -\sin \vartheta_i & 0 \\ 0 & \sin \vartheta_i & \cos \vartheta_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La trasformazione omogenea che permette di individuare la posizione e l'orientazione della terna cinque rispetto alla terna zero sarà quindi la seguente:

$$\hat{T}_5^0(q) = \hat{A}_1^0 \hat{A}_2^1 \hat{A}_3^2 \hat{A}_4^3 \hat{A}_5^4$$

dove le variabili di giunto q_i provengono direttamente dal software di controllo e corrispondono ai valori degli encoder che vengono prima “congelati” da un latch SW ad ogni salita del segnale trigger e successivamente inviati tramite comunicazione ADS. Infine, date le matrici di trasformazione \hat{T}_0^b (base rispetto alla prima terna) e \hat{T}_e^5 (flangia - end effector), è possibile calcolare la cinematica diretta come:

$$\hat{T}_e^b = \hat{T}_0^b \hat{T}_4^0 \hat{T}_e^5$$

Ciò permette di conoscere la posizione e l'orientazione dell'end-effector rispetto al sistema di riferimento base e quindi di ricostruire correttamente le varie lame di scansione nello spazio di lavoro.

3.1.2 End-effector

All'interno di questa sezione è possibile settare i tool offset per ogni tipologia di end effector che si desidera usare. Per i collaudi dimensionale vengono usate da Axist tre tipologie di tool di misura:

- Tastatore
- Snapshot
- Profilometro Laser

A queste si aggiunge la configurazione flangia usata in fase di test ed alla quale non corrisponde nessun offset. L'ultima tipologia di tool, ovvero il profilometro laser, è quella usata sull'ARS per effettuare le operazioni di test sulla misurazione e sulla cinematica. Questa scelta è dovuta principalmente al fatto che il profilometro laser riesce a sfruttare al meglio le potenzialità del sistema senza aver bisogno di una architettura di controllo PLC eccessivamente complessa. Se si pensa infatti all'utilizzo di un tastatore, bisogna tenere in considerazione la gestione della movimentazione di avvicinamento del rubino terminale fino al contatto con la superficie del componente (operazione delicata in cui qualsiasi urto accidentale può provocare il danneggiamento o la rottura del tool) mentre per lo snapshot occorre un posizionamento del tool di misura estremamente preciso rispetto al componente in modo da matchare con certezza i vari “scatti”. Tuttavia, non si esclude la possibilità di usare gli altri tool di misura a seguito di opportune modifiche o integrazione di sistemi di visione e di sistemi di sicurezza software e hardware.

In figura 3.8 viene proposto un esempio di configurazione. Si nota come sia possibile:

- Inserire gli offset del tool rispetto alla flangia di montaggio;
- Inserire il sensor ID del profilometro in uso in modo da consentire ad AVRIS di accedere alle impostazioni del profilometro tramite cavo LAN;
- Indicare la tipologia di scansione e attivare il job desiderato.

Workcell configuration: C:\Users\axs\Documents\Axist\Avris\4.1\ARS\Data\Workcell.xml

File

Manipulators End Effectors Tracking Systems Robotic Systems

End Effector: LMI [ADD] [REMOVE]

Brand: LMI Model: Gocator

Tool Offset:

X:	22.000	α [°]:	90.000
Y:	115.000	β [°]:	180.000
Z:	0.000	γ [°]:	90.000

Angles Convention: EulerXYZ [Quat]

Sensor ID: 76978

Configuration

Feature: HOLE Job Index: 0

Acceleration enabled

Acceleration Parameters

Base Port:	3190	Health Port:	3191
Public Data Port:	3194	Private Data Port:	3195
Web Port:	8081	Upgrade Port:	3192

LaserRefreshRate [ms]: 10

Surface Scan Mode: MULTIPLE_PROFILE

Figura 3.8 – End-Effector, workcell di Avris

3.1.3 Tracking System

All'interno del Robotic System è inoltre possibile configurare uno o più sistemi di tracciamento esterno. Tali sistemi, come si vedrà più ampiamente in seguito, sono essenziali per effettuare una corretta calibrazione e/o mappatura del sistema di scansione. Il metodo di misura del tracker sfrutta un riflettore (una piccola sfera posizionata manualmente dall'operatore sul punto da rilevare) e avviene per "inseguimento" ottico. Così facendo, una volta posizionato il riflettore nella corretta posizione e passato un set di posizioni target, è possibile confrontare la posizione target nominale con la posizione target effettivamente raggiunta. L'insieme di questi dati specifici per ogni asse consente poi di effettuare una calibrazione dell'intero sistema e l'individuazione degli effettivi assi dei giunti.

3.2 Implementazione del modello cinematico nel file master di SA

La necessità di creare un file SA master nasce dall'esigenza progettuale di avere una base di partenza che esuli dal tipo di componente da misurare e dal piano di scansione da applicare. A questa esigenza, si aggiunge anche il vantaggio di aver "standardizzato" il primo step del processo, in quanto, l'operatore dovrà sempre aprire il master posizionato all'interno del Workfolder_ARS e cliccare sull'apposito comando "scriptARS" per poter creare il file di lavoro⁽¹⁾.

⁽¹⁾vedere cap 2, paragrafo 2. qualcosa >>

Vengono adesso elencati i passaggi effettuati per la realizzazione del file master:

1. Creazione della collection (cartella) Robot. Il nome deve corrispondere a quello inserito nel Robot System di AVRIS;
2. All'interno della collection vanno inseriti alcuni elementi necessari all'importazione delle nuvole di punti: un instruments di misura generico e la frame base relativa al sistema di riferimento rispetto alla quale verrà importata la nuvola di punti;
3. Inserire il file di calibrazione in Calibrations e denominarlo "avris". Tale file risulta sempre necessario per AVRIS e permette di importare su SA i parametri calibrati della cinematica e offset tool che andranno a sovrascrivere quelli nominali inseriti nel master. Ciò consente di ottimizzare la simulazione online e offline.
4. Definizione del Robots and Machines nel seguente modo:
 - Per ogni asse del sistema creare il corrispettivo components
 - Ad ogni components associare il modello matematico del link
 - Inserire i components [Base] e [Tool] all'inizio e alla fine della catena

Quanto appena descritto è rappresentato in figura 3.9 in cui viene riportato l'albero di SA del sistema ARS (a) paragonato a quello fino ad ora usato per il Robot AVRIS (b).

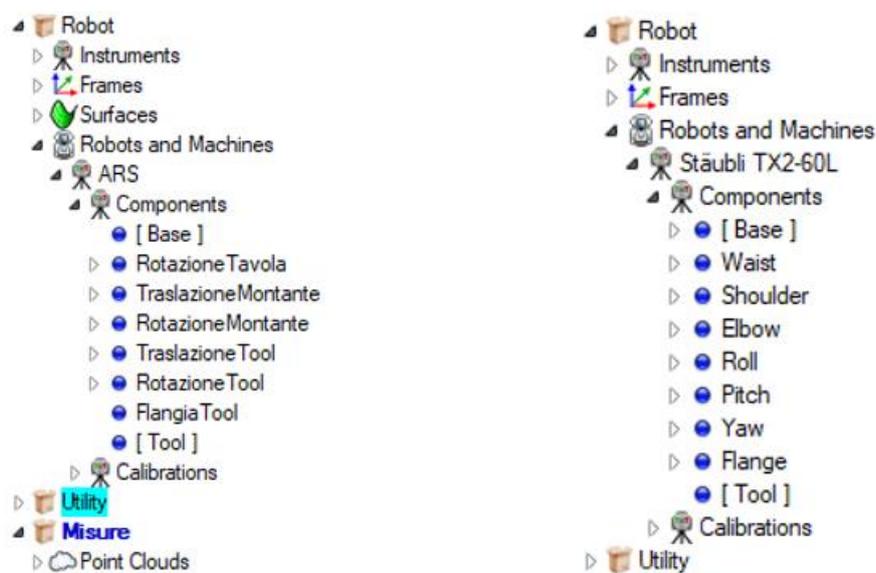


Figura 3.9 : (a) configurazione del sistema ARS, (b) configurazione usata per robot Avris

Nel momento in cui viene creato un components vanno inseriti, tramite l'apposita interfaccia di SA visibile in figura 3.10, i parametri cinematici dei corrispettivi link. La

convenzione sui parametri di *Denavit Hartenberg* usata da SpatialAnalyzer differisce da quella usata in AVRIS (convenzione standard). Partendo quindi dai parametri inseriti nei giunti J1, [...], J6 della workcell di AVRIS si procede nel seguente modo:

- Il parametro A_i (AVRIS) diventa il parametro A_{i+1} (SA)
- Il parametro α_i (AVRIS) diventa il parametro α_{i+1} (SA)
- Il parametro D_i (AVRIS) corrisponde al parametro D_i (SA)
- Il parametro ϑ_i (AVRIS) corrisponde al parametro ϑ_i (SA)

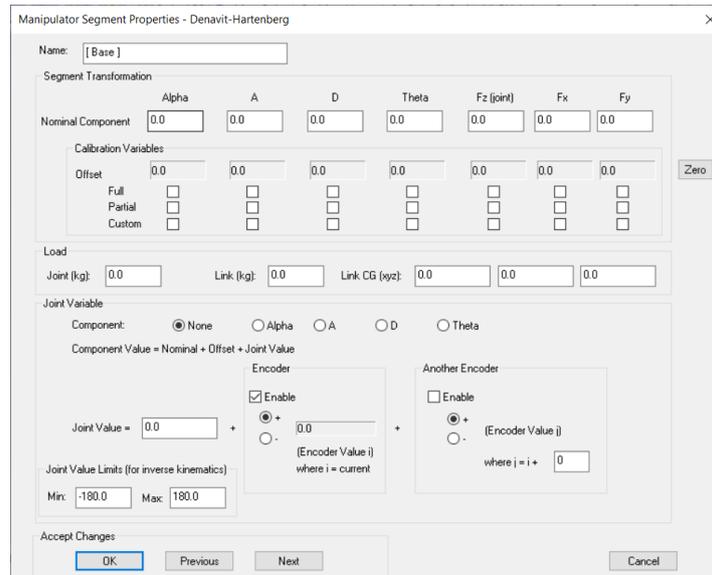


Figura 3.10: interfaccia SA per l'inserimento dei parametri cinematici di DH

Una volta terminata la configurazione è possibile visualizzare le terne di riferimento nello spazio di lavoro e controllare, tramite il teach pendant interno, se la movimentazione degli assi è conforme a quella prevista, figura 3.11:

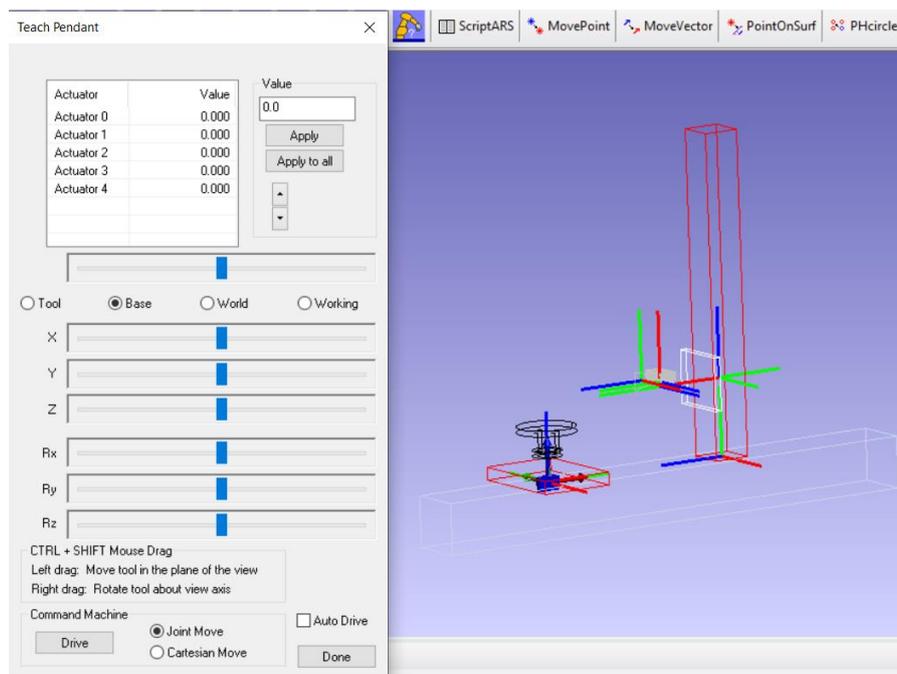


Figura 3.11: Frame nominali dei rispettivi componenti e Teach Pendant interno

3.3 Configurazione temporanea del quinto giunto

Durante la fase di modellazione della catena cinematica si è scelto di prendere come riferimento la versione completa del sistema ARS, ovvero quella caratterizzata dai tre giunti rotoidali e dai due assi prismatici. Nella realtà, al momento dello sviluppo del progetto, l'ultimo giunto per la rotazione del tool di misura nel piano ZX del sistema base non era presente. Si è scelto quindi di implementare una configurazione temporanea che permettesse di effettuare il montaggio del tool in flangia e di effettuare quindi i primi test per poter validare il modello cinematico, per verificare la corretta comunicazione dei software usati e per testare le varie fasi di movimentazione del ciclo macchina. In figura 3.12 viene proposta un'immagine del sistema nel quale sono presenti il giunto prismatico del carrello verticale (1) che consente la traslazione del tool lungo l'asse Z, la configurazione provvisoria staffa-flangia tool (2) montata in macchina e il profilometro laser (3).

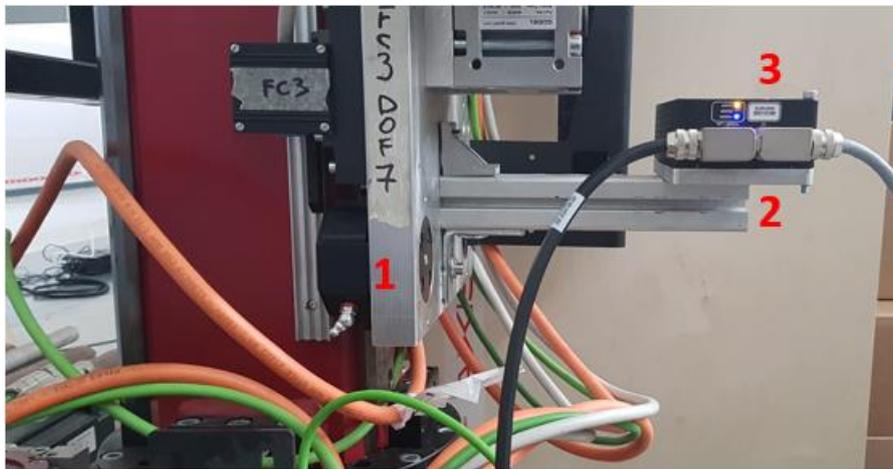


Figura 3.12: configurazione provvisoria J5

È facile notare le differenze tra la configurazione temporaneamente allestita con quella del modello cinematico utilizzato sia in AVRIS che in SA:

- È assente il servo azionamento per la rotazione del tool;
- La struttura meccanica è notevolmente semplificata non essendo necessario garantire la rotazione della flangia rispetto alla flangia.

Questa semplificazione meccanica va ad influire i parametri inseriti nella workcell di AVRIS ed utilizzati nel modello cinematico in fase di test. In particolare, si ha che la variabile di giunto J5 sarà costante ed uguale a zero non essendoci l'effettivo grado di libertà, l'offset fra asse montante e asse J5 (a4) diventa un offset tra asse montante e fine flangia, l'offset tra asse J5 e centro flangia (a5) diventa tra la superficie superiore della staffa e quella superiore della flangia. Nel paragrafo dedicato ai possibili sviluppi futuri verranno elencate le possibili soluzioni costruttive per quanto riguarda la scelta dell'azionamento, del controllo e della macchina dell'ultimo giunto.

3.4 Individuazione dei parametri cinematici calibrati

Una volta definito il modello cinematico del sistema ARS, si è reso necessario calcolare gli effettivi parametri di Denavit-Hartenberg di ciascuno dei cinque giunti che compongono la catena cinematica. Avere infatti l'effettiva posizione reciproca tra i vari giunti nello spazio e con la terna di riferimento base, consente di applicare all'interno di AVRIS la cinematica calibrata alla nuvola di punti che viene generata dal profilometro durante la scansione. Gli strumenti usati per effettuare la calibrazione sono:

- il software di collaudo SpatialAnalyzer;
- un laser tracker modello Leica absolute tracker AT402;
- il software dello strumento "Tracker Pilot".

Il programma Tracker Pilot viene utilizzato per la configurazione iniziale del laser tracker, detta anche fase di inizializzazione. Questo permette di andare a compensare e configurare correttamente lo strumento prima che esso venga connesso a SA. Successivamente, è necessario anche per eseguire controlli e ulteriori calibrazioni se la macchina viene ad esempio smontata o ricollocata. Il Laser Tracker è uno strumento di misura portatile basato su raggio laser che consente di effettuare misure di precisione all'interno di volumi estesi. Generalmente, il Tracker può misurare le coordinate tridimensionali in tre differenti modi: per mezzo di un target (una piccola sfera riflettente che viene posizionata manualmente dall'operatore sul punto da rilevare), per mezzo di un tastatore senza fili punto a punto e per mezzo di uno scanner laser ad alta velocità. Il metodo di misura da utilizzare varia in funzione del tipo di applicazione ed in tutti i casi avviene per "inseguimento" ottico del tool di misura da parte del laser. L'applicazione di questi sistemi si è diffusa durante il corso degli anni e ormai spazia dal collaudo dimensionale al reverse engineering, passando per le misurazione di rilievi e di riposizionamento pezzi.

Lo strumento di misura usato è un laser tracker AT402 prodotto dalla Hexagon, leader mondiale nel settore dei sensori e dei software di misura, figura 3.12 (a). Tra le caratteristiche principali dello strumento vi sono:

- Esteso campo di temperatura d'esercizio che va dai -15 a +45 gradi Celsius;
- Volume di misura fino a 320 metri (\emptyset), con distanza di misura minima inferiore a 0,8 metri;
- Funzione di power lock, il raggio interno, se interrotto, si aggancia automatica senza alcun intervento dell'operatore;
- Funzione orient to gravity, consente di misurare avendo allineato l'asse Z al vettore gravità.

A questi tipi di strumenti sono associati dei massimi errori permessi, detti MPE, definiti nell'ISO-VIM (International Vocabulary of basic and general terms in Metrology). Il MPE può essere considerato come il valore dell'errore massimo garantito per tutte le misurazioni effettuate con quel particolare strumento. Nello specifico dell'AT402 si ha una accuratezza garantita della misurazione nel centesimo di metro. Durante la fase di calibrazione sono stati usati dei riflettori ottici, detti target, costituiti da un corpo esterno sferico di acciaio inossidabile, temprato e sferico. All'interno è invece presente il

comparto ottico che consente un'elevata accuratezza di agganciamento del laser e di riflessione, figura 3.13 (b). (Faro, 2020)

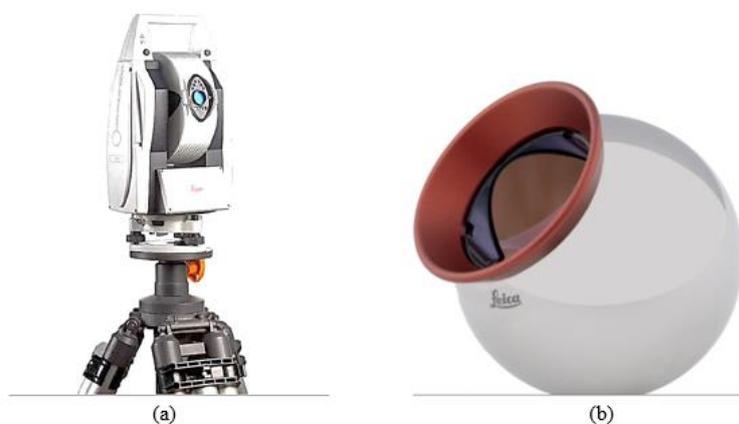


Figura 3.13: (a) Leica Absolute Tracker AT402, (b) Red Ring Reflector Leica

Per rendere ripetibile il processo di calibrazione e allineamento del laser rispetto al sistema ARS, sono stati installati sei porta target in sei posizioni fisse, figura 3.13. Prima di effettuare le operazione di calibrazione si procede al posizionamento del riflettore in ciascuno dei porta target in modo da acquisire i punti nominali di allineamento. Tale procedura, permette di vincolare i sei gradi di libertà posseduti dal laser tracker nello spazio e consente di riprendere la misurazione senza introdurre errori anche nel caso in cui si dovesse spostare lo strumento di misura. Infatti, basterebbe semplicemente riacquisire i sei punti in corrispondenza dei porta target ed effettuare un best fitting rispetto ai valori nominali. Come risultato si avrà la rototraslazione delle misurazioni nelle corrette posizioni.

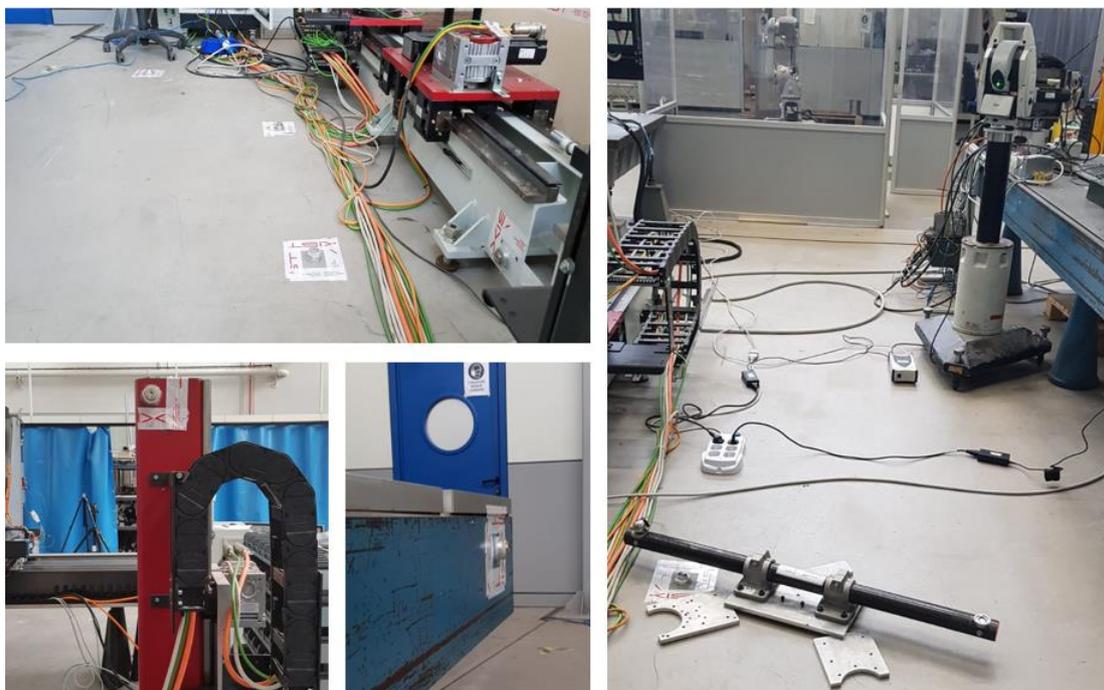


Figura 3.14: posizionamento dei porta target nei pressi dello strumento di misura, laser e scale bar di taratura

Come primo step per l'attività di calibrazione è stata eseguita una procedura di inizializzazione del laser tracker. Questa si rende necessaria ed obbligatoria in qualsiasi attività di misurazione, in quanto ad ogni riallocazione dello strumento potrebbe verificarsi una staratura dello stesso o, nel peggiore dei casi, un danneggiamento. La procedura prevede una messa in bolla del laser tramite tre apposite rotelle che controllano l'inclinazione laterale e trasversale dello strumento rispetto alla verticale (calcolata in corrispondenza del vettore gravità). Fatto ciò, si accede al programma Tracker Pilot e si collega il laser tramite l'indirizzo IP dello stesso. Una volta configurato anche il tipo di riflettore usato, lo si aggancia al fascio laser e lo si ricolloca su uno dei porta target. A questo punto viene effettuata in automatico dal sistema un'operazione di field check che prevede una serie di movimentazioni della testa del laser alternate ad agganciamenti del riflettore. Finita l'operazione, bisogna invece effettuare manualmente lo scale bar check che consiste nel misurare una scale bar, precedentemente settata all'interno del sistema, sia longitudinalmente che trasversalmente. Per fare ciò, occorre posizionare il target nei due appositi alloggiamenti ricavati all'interno del tubolare in carbonio, come visibile in figura 3.15. Ciò consente di verificare l'accuratezza di misurazione dello strumento, andando a confrontare il valore nominale della scale bar (scalato in funzione della temperatura ambiente) con quello misurato dal laser.

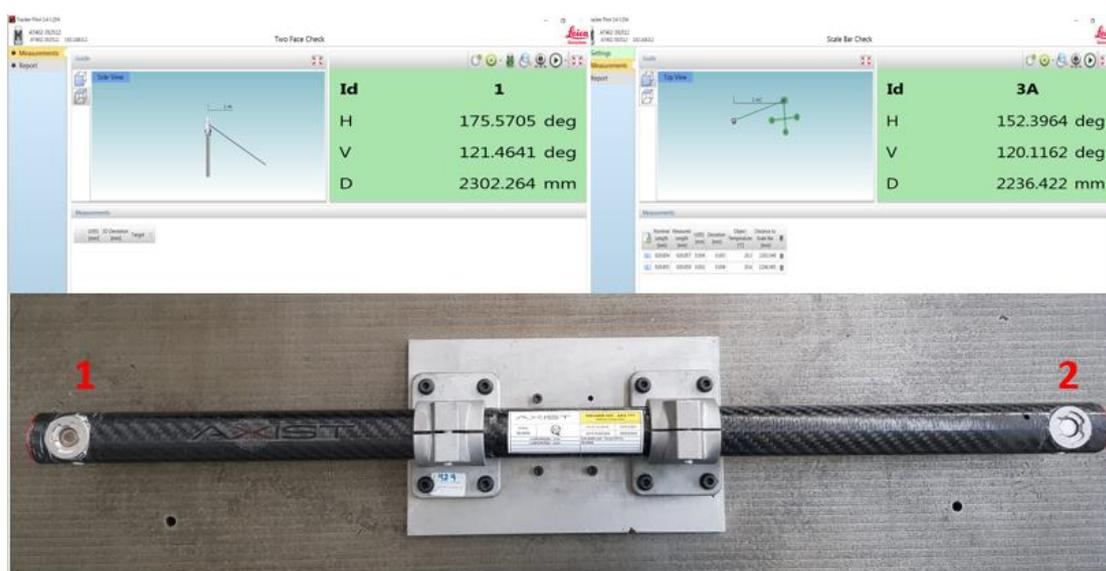


Figura 3.15: Field check, scale bar check e barra di calibrazione Axist AXS777

Terminata la procedura di inizializzazione del laser si è potuto procedere con le rilevazioni atte alla determinazione dei parametri cinematici calibrati del sistema ARS. Per allineare lo strumento al CAD, presente nel file SA, è stata valutata l'effettiva terna di riferimento base della catene cinematica tramite la misurazione delle seguenti entità geometriche:

- Piano XY, preso lungo la guida che consente lo scorrimento orizzontale del carrello tavola;
- Asse Z, preso come l'asse di rotazione della tavola quando il carrello orizzontale si trova nel fincorsa;
- Asse X, preso lungo la direzione di traslazione della tavola.

La valutazione delle entità geometriche viene effettuata in maniera automatica in SA andando a definire in anticipo delle Relationships dello stesso tipo. Ad esempio, nel caso della rilevazione dell'asse z_0 è stato necessario acquisire l'effettiva rotazione del piano tavola. Su SA è stata quindi definita, all'interno della collezione "SistemaBase", una relationship di tipo cerchio e attivata durante la fase di acquisizione mentre il target posizionato sul piano tavola seguiva la rotazione della stessa in modo solidale. Tramite l'interazione tra interfaccia strumento e piano di misura su SA, i punti così acquisiti vengono direttamente usati per fittare dal cerchio. Una volta ottenuto si è poi costruito l'effettivo asse di rotazione della tavola, considerando la normale al cerchio passante per il suo centro. Una procedura molto simili è stata seguita per la determinazione delle altre due entità geometriche della terna base, andando a definire relationships di tipo piano e linea. Costruita quindi l'effettivo sistema di riferimento base, si è poi allineato lo strumento e i punti misurati al CAD.

In maniera del tutto analoga sono poi state valutate le altre terne, riferite ai cinque giunti del sistema. Particolare attenzione è stata data alla terna del giunto J3 in modo da verificare l'effettiva inclinazione dell'asse di rotazione del montante con il piano delle guide. Una volta fissato il porta target sul carrello verticale, sono state effettuate diverse rotazioni del montante andando ad incrementare la quota del target. Durante le varie rotazioni sono stati acquisiti i punti e fittati i vari cerchi. Successivamente, dai diversi cerchi sono stati poi estrapolati i vari centri di rotazione ed applicando un best fitting è stato infine possibile determinare l'asse di rotazione montante (z_2) e da lì la restante terna, figura 3.16.

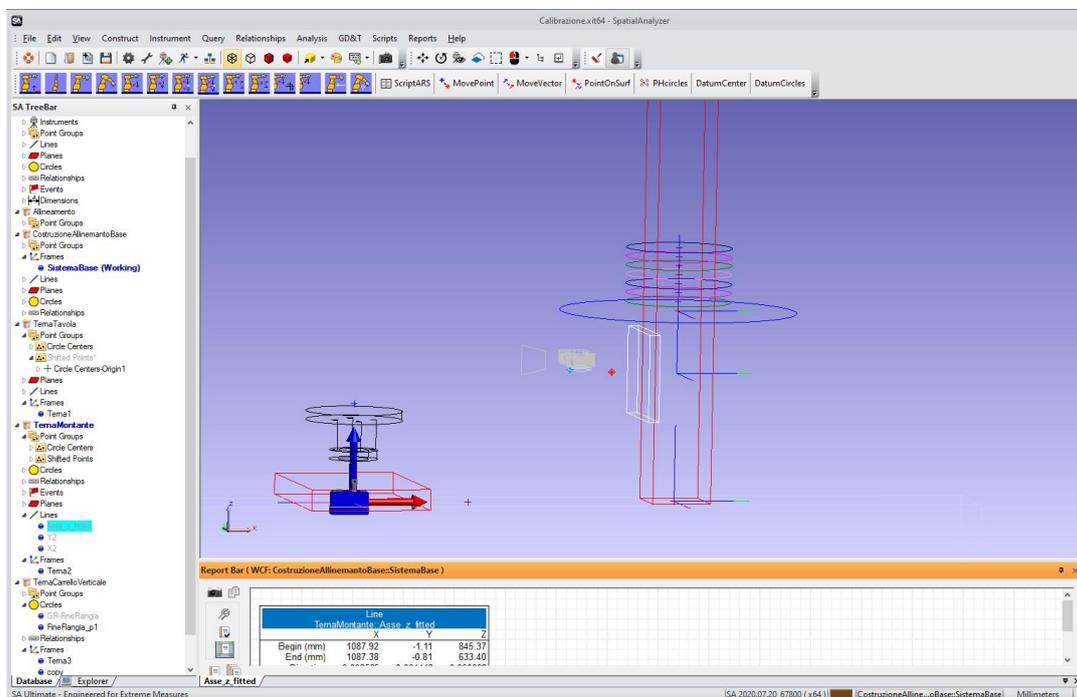


Figura 3.16: processo di best fitting per la determinazione dell'asse di rotazione montante

Sfruttando poi una serie di funzionalità di SA sono stati trovati i valori delle rotazioni e degli offset tra le varie del catene cinematica. I valori ottenuti sperimentalmente durante la calibrazione vengono proposti nella seguente tabella.

	a_i	d_i	α_i	ϑ_i
0-1	0	0	90°	-90° + q ₁
1-2	-0.09	-1085.75 + q ₂	-90.147°	-0.082°
2-3	0	426.42	0	q ₃
3-4	390.932	q ₄	-90°	-90°
4-5	9.26	0	0	-90° + q ₅

Tabella 3.3: parametri di DH risultanti dall'attività di calibrazione

3.5 Analogie con altri strumenti di misura e bracci robotici

Il sistema ARS, in particolare il sottosistema montante, presenta diverse analogie con altri sistemi alcuni dei quali usati proprio nell'ambito del collaudo dimensionale. Un esempio sono le CMM a braccio orizzontale che presentano caratteristiche di funzionamento molto simili a quelle del sottosistema montante. In figura vengono riportati due modelli Pro T Premium della Zeiss, azienda leader nella fornitura di macchine di misura e software di collaudo, montanti in parallelo direttamente sulla linea di produzione.

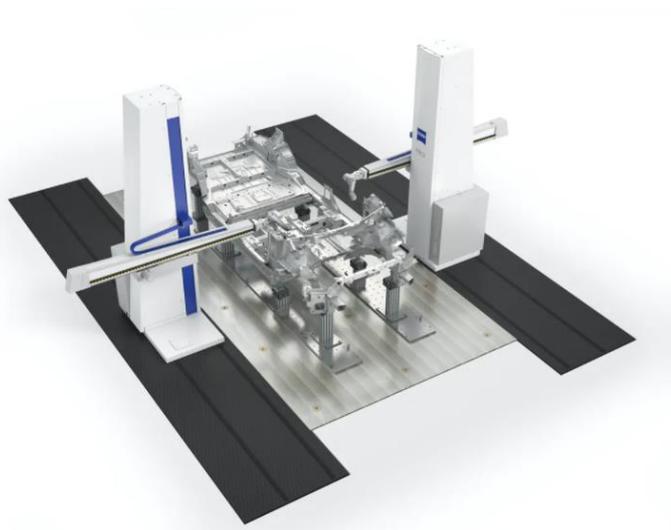


Figura . 3.17: Zeiss Pro T Premium, CMM a braccio orizzontale

Si può notare come entrambi i sistemi abbiano in comune la traslazione orizzontale del montante e la traslazione verticale del tool di misura. Altra analogia può essere riscontrata se si considerano tre tipologie di manipolatori ampiamente usati in ambito industriale: il manipolatore cartesiano, il manipolatore cilindrico ed il manipolatore sferico. (Zeiss, 2020)

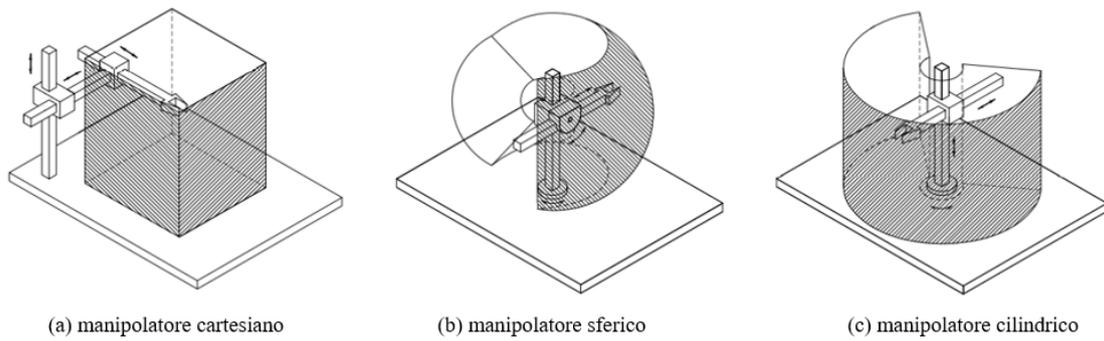


Figura 3.18: topologie di manipolatori usati in ambito industriale

Del manipolatore cartesiano si riprende la rigidezza complessiva del sistema, data principalmente dalle guide lineari Gudel e dalla trasmissione del moto tramite accoppiamento pignone dentiera. Del manipolatore cilindrico viene ripreso il giunto prismatico, mentre per quanto riguarda i giunti rotoidale si nota la similitudine con il polso sferico. La precisione di posizione del tool di misura sarà anche nel caso dell'ARS influenzata dallo sbraccio radiale dovuto all'ultimo giunto rotoidale.

4. Gestione della rotazione del montante

Nel capitolo precedente sono stati presentati e analizzati i cinque gradi di libertà che compongono il sistema ARS. I primi due, ovvero la rotazione e traslazione della tavola, consentono di avvicinare il componente da misurare al tool di misura e di effettuare una rotazione completa dello stesso in fase di scansione. Queste movimentazioni, al pari della traslazione del tool lungo Z, saranno sempre presenti a prescindere dal tipo di componente da scansione. Caso diverso si ha per la rotazione del sottosistema montante, la quale si rende necessaria nel momento in cui l'angolo di incidenza, che corrisponde all'angolo tra il fascio laser e la superficie del componente, supera un certo angolo limite. Questo angolo limite è dettato da una serie di limiti operativi del tool di misura usato in fase di test e per il quale il sistema è principalmente predisposto, ovvero il profilometro laser LMI, tipologia 2430. Nella realtà applicativa, si ha un vero e proprio cono di ammissibilità di scansione che può essere scomposto in due angoli limite:

- Theta, che viene gestito dalla rotazione del montante;
- Omega, che viene gestito dall'ultimo asse, attualmente non implementato, che consente la rotazione rispetto all'asse X (sistema di riferimento base) del solo tool di misura.

Nel seguente capitolo verrà affrontata la sola correzione da apportare in seguito alla rotazione del montante in quanto il quinto giunto avente lo scopo di ruotare direttamente il tool in flangia è ancora in fase di sviluppo. L'informazione sull'angolo limite è stata presa direttamente dal catalogo LMI, del quale si riportano le immagini di esempio di figura 4.1 e poi verificata tramite appositi test.

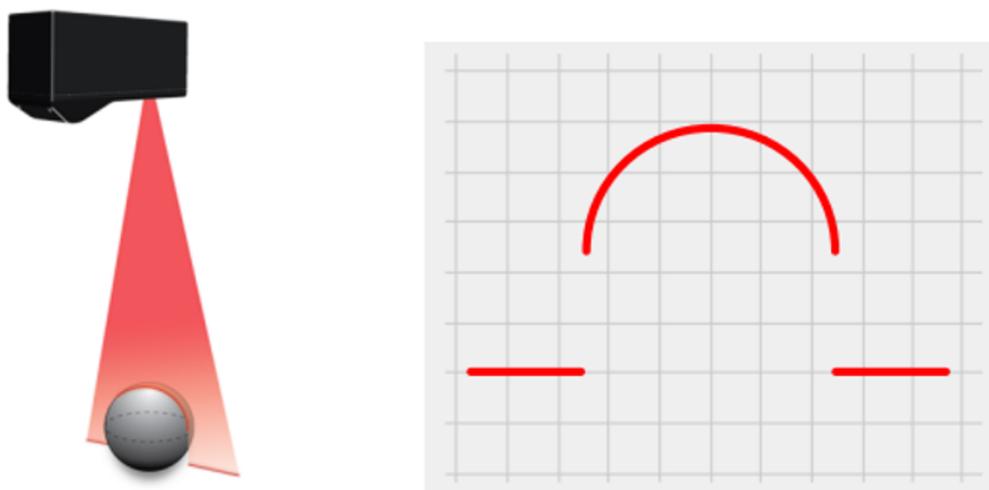


Figura 4.1: angoli di incidenza limite durante la scansione, Gocator LMI 2430

Come si nota dalla figura precedente, l'acquisizione risulta essere soddisfacente fino a quando non vengono raggiunti angoli di incidenza prossimi e superiori ai 90° . Nell'intorno di tale angolo la scansione risulta essere compromessa in quanto il raggio riflesso non viene catturato dalla cellula ricevente montata nel profilometro e non

consentendo, a sua volta, di effettuare la triangolazione laser per il calcolo della distanza tra profilometro e superficie. (LMI, 2020)

4.1 Modello cinematico e di calcolo

Con lo scopo di semplificare il calcolo della corretta rotazione del montante, è stato sviluppato un modello matematico basato sul funzionamento del sistema biella-manovella semplice, ovvero caratterizzato da eccentricità nulla. Per meglio capire la trattazione presentata nei prossimi paragrafi, risulta utile specificare nel dettaglio la nomenclatura che verrà usata da qui in avanti e che è già possibile osservare in figura 4.2:

- *Raggio di manovella* (m), tramite il quale si fa riferimento alla distanza fra il centro di rotazione della tavola ed il punto di applicazione del vettore uscente sulla superficie del pezzo \vec{v} . Tale valore corrisponde alla coordinata lungo l'asse X del punto di applicazione del vettore ruotato (vedere paragrafo dedicato, capitolo 2);
- *Lunghezza di biella* (b), tramite la quale si fa riferimento alla distanza fra il centro di rotazione del montante ed il centro del trapezio di scansione. Tale valore resta costante e pari a zero, in quanto l'angolo di alzata facente riferimento al quinto ed ultimo giunto non viene per il momento controllato;
- *Spostamento del piede di biella* (x_{manov}), si traduce nella correzione da passare al secondo giunto che controlla la traslazione della tavola rotante.

Angoli:

- *Alpha* (α), rotazione da applicare al montante. Nel modello corrisponde all'angolo di biella;
- *Theta* (ϑ), angolo da tra vettore uscente e raggio laser (angolo da ridurre);
- *Gamma* (γ), complementare di theta quando manovella e biella sono allineati ovvero è l'angolo tra il raggio di manovella m ed il vettore uscente \vec{v} in B. Questo angolo resta costante anche dopo la rotazione poiché manovella e vettore uscente ruotano solidali;
- *Beta* (β), rotazione compiuta dal punto B per portarsi in B' rispetto al centro tavola, a seguito della correzione su alpha. Tale angolo corrisponde nel modello all'angolo di manovella;
- *Theta primo* (ϑ'), nuovo angolo tra vettore uscente e raggio laser a seguito della correzione alpha.

Nella prima immagine (a) di figura 4.2 è rappresentata la condizione limite in cui il vettore, che fa riferimento alla normale uscente di un generico punto B sulla superficie del pezzo, è ortogonale al fascio laser di scansione ($\vartheta = 90^\circ$). In questa configurazione, il montante non è ancora ruotato ($\alpha = 0^\circ$) e quindi il fascio laser passerebbe idealmente dal centro di rotazione della tavola rotante.

Si può facilmente notare come questa particolare configurazione iniziale possa essere equiparata ad un sistema biella-manovella nella posizione di punto morto superiore. Inoltre, tale configurazione sarà comune a tutti i vettori in quanto all'interno della function di primo livello si effettua una rotazione di tutti i vettori in modo da portare il punto di applicazione (B) lungo l'asse X del sistema di riferimento base.

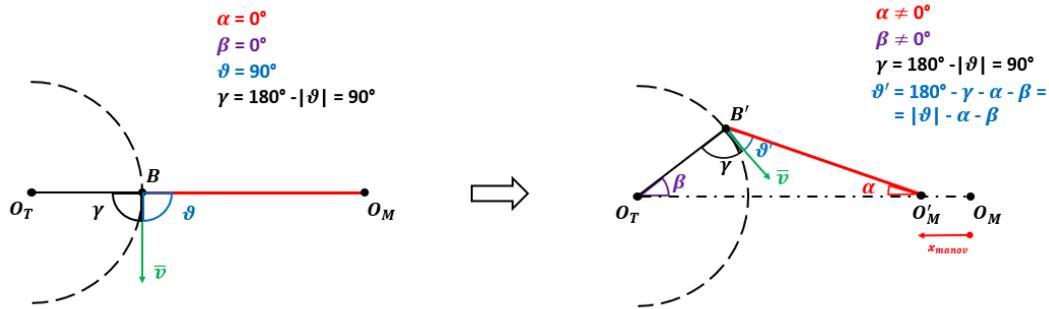


Figura 4.2: modello cinematico, (a) configurazione nominale e (b) configurazione con rotazione montante

Se si impone la condizione che il centro del trapezio di scansione debba sempre coincidere con il punto B e si applica una certa rotazione α , risulta abbastanza evidente l'analogia con la cinematica di un manovellismo centrato, immagine (b) di figura 4.2. Infatti, per rispettare la precedente condizione, il centro di rotazione del montante O_M dovrà spostarsi verso il centro tavola della quantità x_{manov} , proprio come farebbe un piede di biella. Questa correzione lungo l'asse X verrà applicata in moto relativo all'asse che controlla la traslazione della tavola in quanto il montante non può traslare ma solo ruotare, ovvero sarà la tavola ad avvicinarsi e non viceversa. Identificato quindi il modello cinematico di riferimento, occorre trovare una procedura che permetta di verificare se il nuovo angolo ϑ' , generato a seguito della rotazione α , sia effettivamente diminuito.

Facendo riferimento alla seconda immagine di figura 4.2, si consideri il triangolo avente come vertici i punti $O_M B' O_T$. L'angolo ϑ' può essere espresso come:

$$\vartheta' = 180^\circ - \alpha - \beta - \gamma \quad (1)$$

Si ricorda dalle definizioni precedenti che gamma è l'angolo tra il vettore uscente ed il raggio di manovella. Esso presenterà un valore che resta costante per qualsiasi rotazione α del montante, in quanto il punto B e il raggio di manovella $m(\overline{O_M B'})$ ruotano solidali. La (1) si può riscrivere come:

$$\vartheta' = \vartheta - \alpha - \beta \quad (2)$$

$$\gamma = 180^\circ - |\vartheta|$$

dove per definizione si è sostituito a gamma.

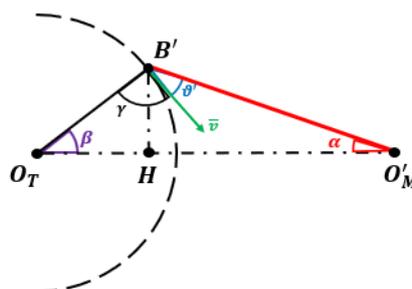


Figura 4.3: costruzione geometrica

Segue, inoltre, che α e β essendo rispettivamente gli angoli di biella e manovella non sono tra loro indipendenti. Facendo adesso riferimento ai triangoli rettangoli $B'H O_M$ e $B'H O_T$ visibili in figura 4.3, è possibile esprimere il cateto comune come:

$$B'H = b * \sin \alpha$$

$$B'H = m * \sin \beta$$

Esplicitando adesso l'angolo β :

$$\beta(\alpha) = \sin^{-1} \left(\frac{b}{m} * \sin \alpha \right)$$

La relazione che consente di calcolare ϑ' , una volta imposto l'angolo di rotazione del montante, è costituita quindi da valori noti (α e γ) e valori funzione di alpha (β) e può essere riscritta nelle due seguenti forme:

$$\vartheta' = 180^\circ - \alpha - \beta(\alpha) - \gamma$$

$$\vartheta' = \vartheta - \alpha - \beta(\alpha)$$

Se il ϑ' risultante dal calcolo precedente risulta essere validato, ovvero se

$$\vartheta' \leq \vartheta'_{lim} = 90^\circ$$

è possibile calcolare lo spostamento del piede di biella O_M in funzione dell'angolo $\beta(\alpha)$ e del raggio di manovella m in quanto b , lunghezza di biella, è fisso:

$$x_{manov} = m * \left(1 - \cos \beta + \frac{b}{m} - \sqrt{\left(\frac{b}{m}\right)^2 - \sin^2 \beta} \right)$$

4.2 Integrazione del modello cinematico nello script Matlab

Come si evince dal paragrafo precedente, l'obiettivo ultimo è quello di "abbattere" il valor iniziale di ϑ in modo da giungere ad un nuovo angolo ϑ' e rendere quella specifica zona scansionabile. Nello schema di figura 4.4, viene presentata la logica di calcolo implementata su Matlab.

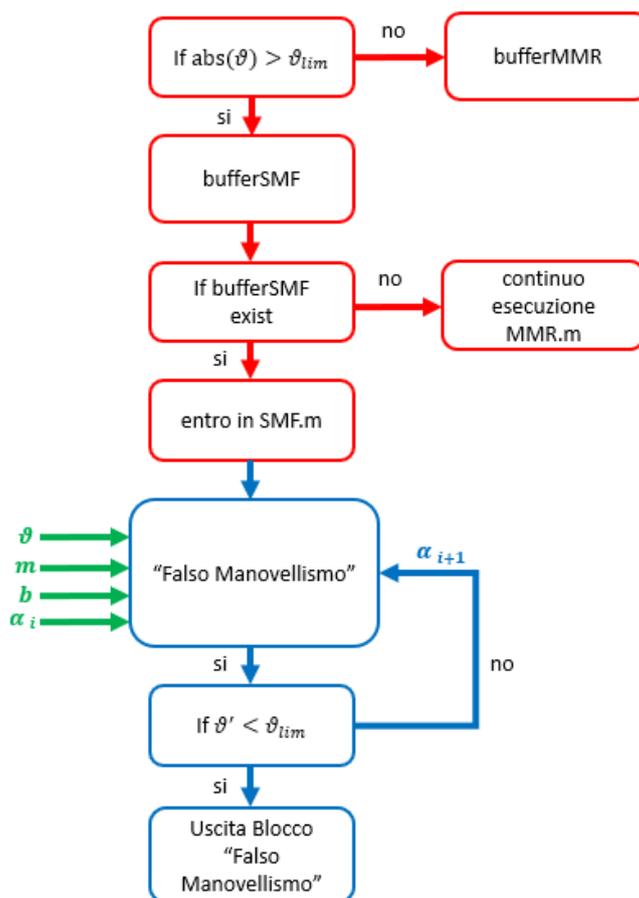


Figura 4.4: logica di calcolo

I blocchetti di colore rosso sono operazioni che si trovano all'interno della function di primo livello "MMR.m" mentre il resto è stato implementato nella function di secondo livello "SMF.m". Il primo IF ha lo scopo di individuare tutti i vettori critici che presentano un angolo ϑ maggiore di quello massimo accettabile dal profilometro. Se il risultato dell'IF è negativo il vettore viene raccolto nel bufferMMR e processato nella function di primo livello "MMR.m", viceversa viene raccolto nel bufferSMF. A questo punto per determinare l'ingresso nella function di secondo livello "SMF.m" si verifica l'esistenza del bufferSMF, in quanto esso viene generato se e solo se almeno un vettore presenta angolo maggiore di quello limite. Una volta che la function di secondo livello viene attivata, si entra nel blocco "Falso Manovellismo" in cui sono implementate le equazioni viste nel paragrafo precedente. Dati in ingresso i valori caratteristici del sistema esse permettono di calcolare:

- il nuovo angolo ϑ' ;
- l'angolo α da passare alla rotazione del montante;
- lo spostamento x_{manov} da passare alla tavola.

Una precisazione va fatta per quanto riguarda l'angolo α_i che in questa fase preliminare è da considerare come una variabile interna al blocco "Falso Manovellismo". Ciò è dovuto al fatto che una volta calcolato l'angolo ϑ' si effettua una verifica sulla validità del risultato ottenuto, andandolo a confrontare con il valore limite. Se il decremento risulta insufficiente ($\vartheta' \geq \vartheta'_{lim}$) si esegue una seconda iterazione con un angolo di biella α_{i+1} che è pari al valore precedente incrementato del passo scelto.

Questo procedimento iterativo continua fino a quando non viene individuato l'angolo ϑ' che verifica l'IF di uscita. Prima di salvare i valori trovati occorre però riportare il corretto segno sull'angolo α in quanto l'angolo ϑ , variabile input, può essere sia positivo che negativo. Per tenerne conto, si effettua la seguente operazione:

$$\alpha = \alpha * \frac{\vartheta}{|\vartheta|}$$

Per lo stesso motivo, quando si procede al calcolo di gamma nel blocco cinematico si considera il valore assoluto di ϑ :

$$\gamma = 180^\circ - |\vartheta|$$

L'intero procedimento fino ad ora esposto si basa sul fatto di avere una serie di valori fissi di α , quindi ad esempio durante l'iterazione si avrà una sequenza del tipo:

$$\alpha_i = 0.5^\circ$$

$$\alpha_{i+1} = 0.5^\circ$$

$$\alpha_{i+2} = 0.5^\circ$$

$$\alpha_{i+3} = 0.5^\circ$$

[...]

data dal ciclo for che si occupa di incrementare il valore di α (in questo caso con un passo pari a 0.5°).

Per poter validare il procedimento di calcolo è stato implementato uno script su Matlab, "DataSetManovellismo.m", con il quale è stato possibile costruire un data set completo dei valori di α e x_{manov} possibili. Lo script si basa su un triplice ciclo for indentato, dove nei due più interni vengono fatti ciclare gli array di m (distanza punto di applicazione vettore e centro tavola) e ϑ (angolo tra vettore uscente e fascio laser) in modo da avere tutte le possibili combinazioni tra di essi. Nel ciclo for più interno si valuta quale sia il parametro di α , tra quelli fissati, che per primo riesce ad abbattere il valore dell'angolo ϑ o, in altri termini, che permette di avere un $\vartheta' \leq \vartheta'_{lim}$. In figura 4.5 ed in figura 4.6, vengono riportati due grafici per capire l'influenza che hanno i parametri m e ϑ sul calcolo di α (e quindi x_{manov}) e l'influenza della scelta sul passo da applicare ad α .

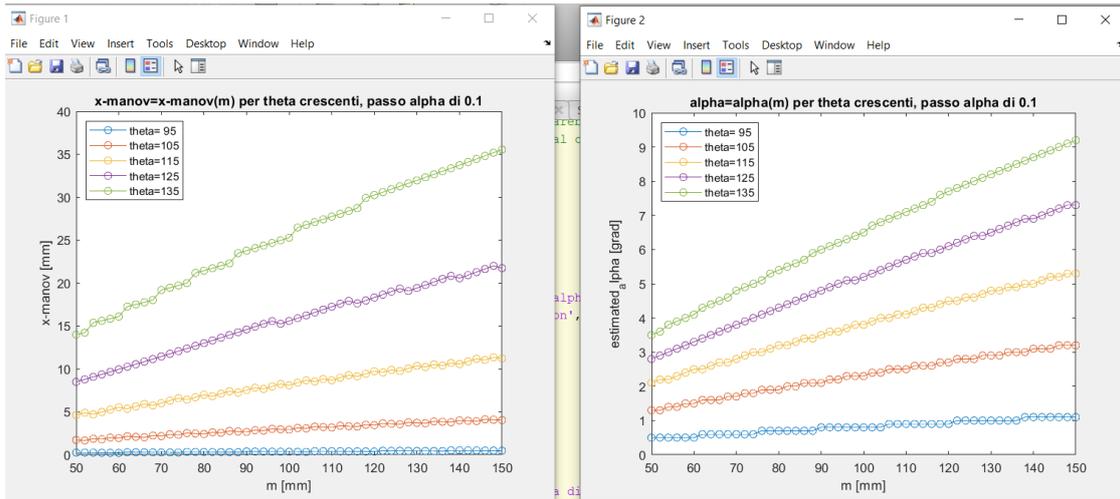


Figura 4.5: data set falso manovellismo con passo di alpha a 0.1

Entrambi gli andamenti sono stati generati a partire dagli stessi valori input di m (range 50:2:150) e ϑ (range 90:1:135). L'unica differenza si ha sul passo scelto per α , in particolare in figura 4.5 vi sono i risultati con passo pari a 0.1° mentre nella figura 4.6 sono presentati i risultati con un passo maggiore e pari a 0.5° .

Si vede come all'aumentare di m e ϑ , o di uno dei due, sia α che x_{manov} subiscono un considerevole aumento. Si vede, inoltre, come per ϑ bassi l'influenza del raggio di manovella vada ad incidere poco sul valore di α e la differenza tra valore iniziale e finale di x_{manov} sulle prime due curve "isotheta" è relativamente contenuta. Per ϑ elevati si hanno scostamenti maggiori tra i valori iniziali e finali della stessa curva. Come ultima analisi, avere la possibilità di far "scattare" α con una frequenza maggiore permette di abbattere più efficacemente tutte le combinazioni di m - ϑ e limitare il fenomeno delle rampe.

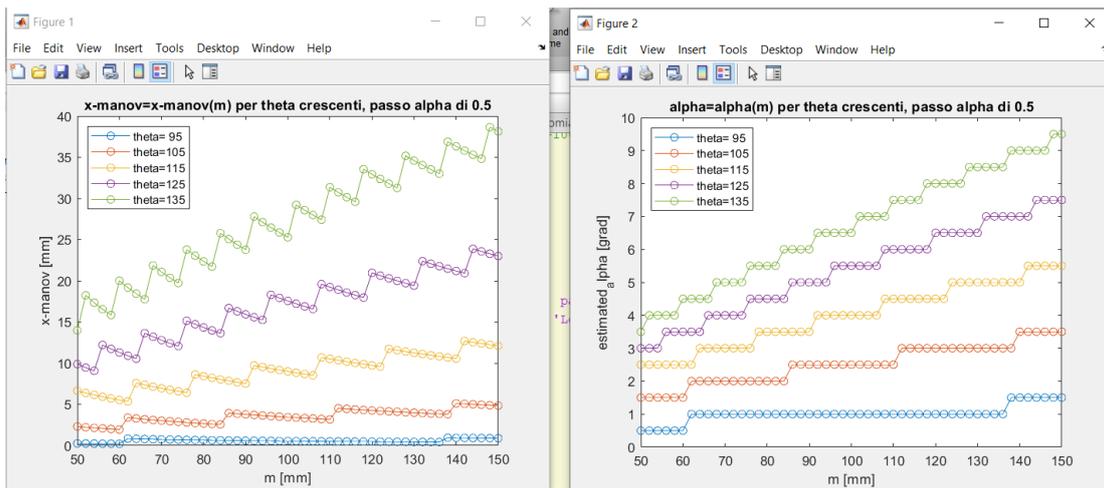


Figura 4.6: data set falso manovellismo con passo di alpha a 0.5

Proprio partendo dall'ultima considerazione, si è proceduto con l'elaborazione di un secondo approccio che permettesse di calcolare i parametri di α e x_{manov} .

Tale approccio non considera più α come una variabile interna ma come variabile di output che potrà quindi assumere qualsiasi valore all'interno del range $\pm \alpha_{max}$.

I parametri di input saranno b , m , ϑ e l'angolo ϑ' , che viene imposto uguale ad un valore ϑ'_{lim} pari o inferiore a 90° (a seconda se si vuole essere più o meno conservativi). Partendo dalle stesse formulazioni ricavate per il modello precedente, si arriva quindi al seguente sistema:

$$\begin{cases} \vartheta' = \vartheta - \alpha - \beta(\alpha) \\ \beta(\alpha) = \sin^{-1}\left(\frac{b}{m} * \sin \alpha\right) \\ \vartheta' = \vartheta'_{lim} \end{cases}$$

Dal quale è possibile esplicitare α come

$$\alpha = \sin^{-1}\left(\frac{\sin(\vartheta - \vartheta'_{lim})}{1 + \frac{b}{m}}\right)$$

Fissato ϑ' ed il valore della lunghezza di biella, si vede come l'angolo α resti funzione dei soli parametri m e ϑ riferiti al punto di applicazione del vettore uscente e al vettore uscente stesso. Anche in questo caso si può pensare di utilizzare il valore assoluto di ϑ , per poi recuperare il segno come visto in precedenza.

In figura 4.7 vengono confrontati i risultati ottenuti imponendo un passo di α pari a 0.1° con quelli ottenuti dal calcolo diretto di α .

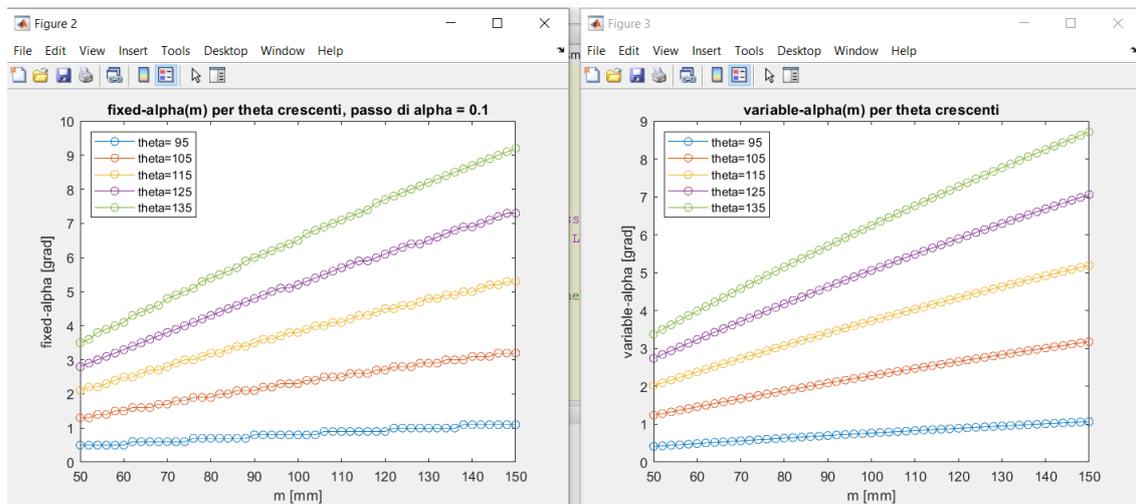


Figura 4.7: a sinistra valori di α con passo pari a 0.1 , a destra valori di α calcolati in un range continuo

Anche se non è più presente l'effetto "rampa", gli andamenti nei due casi risultano essere molto simili. La scelta tra il primo o il secondo approccio dipende da due aspetti fondamentali:

1. Vedere quale approccio garantisce la miglior calibrazione del sistema;
2. Vedere la minima rotazione α che il montante riesce a garantire.

4.3 Considerazioni geometriche

Facendo sempre riferimento al modello cinematico biella - manovella, fissato un certo raggio di manovella, il valore massimo dell'angolo che può essere raggiunto dalla biella è dato da:

$$\alpha_{max} = \sin^{-1}\left(\frac{m}{b}\right)$$

Il valore di α_{max} varia quindi in funzione del raggio di manovella (essendo b un valore fissato e uguale alla distanza centro montante – centro trapezio) e può essere usato per:

- ulteriore equazione di verifica all'interno dell'algoritmo;
- individuare il valore massimo del passo che può essere applicato ad α nell'approccio con α fissati.

Infatti, nel caso in cui il passo di α sia eccessivamente alto, l'algoritmo darebbe in uscita dei valori di α non più compatibili con il movimento simil biella-manovella e di conseguenza non si avrebbe intersezione tra fascio laser e superficie pezzo. In figura 4.8, viene evidenziato proprio il verificarsi di questo fenomeno, paragonabile ad un distacco fisico tra la manovella e la biella, nel momento in cui si utilizza un passo di α pari a 3. In rosso è indicata la retta costituita dagli α_{max} che è infatti una funzione lineare rispetto alla lunghezza di manovella.

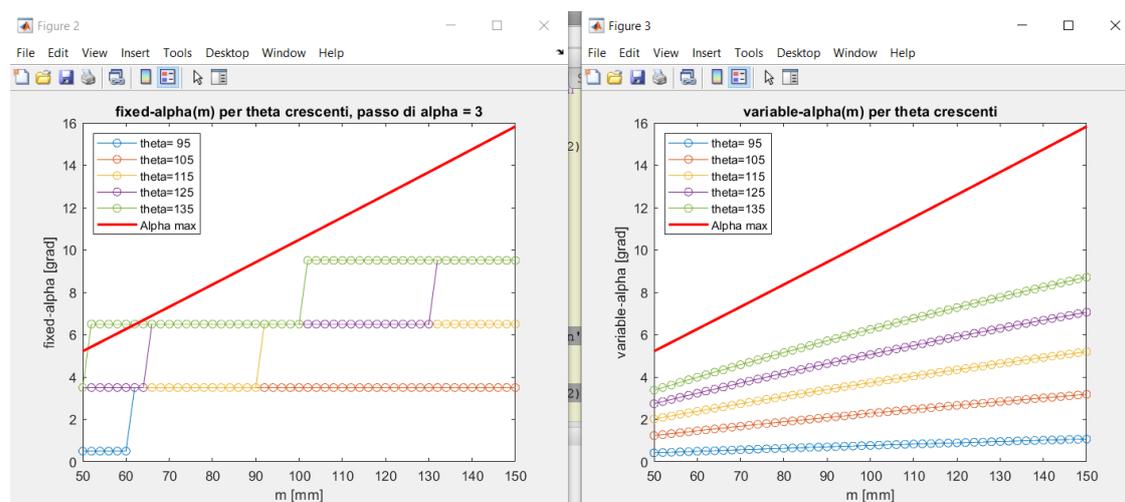


Figura 4.8: fenomeno del "distacco" biella-manovella

Sperimentalmente, una volta assegnato il valore della lunghezza di biella, si è quindi potuto calcolare il limite superiore che si può assegnare al passo senza eccedere il valore di α_{max} su tutto il range di m . Tale valore risulta essere pari a 2.3° . Nell'approccio con α variabili invece non si verifica questa problematica, in quanto il calcolo dell'angolo di biella non richiede nessun incremento.

4.4 Ottimizzazione delle correzioni calcolate per il generico piano di scansione

Tramite il modello si ottengono come valori output, per ogni correzione, la rotazione α da passare all'asse che gestisce la rotazione del montante e la compensazione lungo x_{manov} , l'asse X, da passare all'asse di traslazione della tavola. Tali output saranno gli stessi sia se si usa l'approccio con alpha fissati sia se si sceglie l'approccio con alpha variabili.

A questo punto si rende evidente la necessità di trovare un algoritmo di fitting che permetta di ottimizzare ed accorpare, qualora possibile, tutte le correzioni trovate in un singolo piano di scansione. I parametri da tenere in considerazione per effettuare il fitting saranno quindi le coppie di α e m associate ad ogni correzione calcolata. In particolare, questo algoritmo si rende necessario se all'interno di un piano vengono individuate almeno due coppie diverse. I possibili casi che si possono verificare sono i seguenti:

1. α simili ed m simili (caso ottimale)
2. α diversi ma con m simili (caso di facile ottimizzazione)
3. α diversi con m diversi (difficile ottimizzazione)

La scelta di usare il raggio di manovella come parametro di partenza per effettuare il fitting è legata al fatto che l'area di scansione è stata, fino ad ora, puntualizzata nel centro del trapezio di scansione stesso. Inoltre, il parametro x_{manov} è funzione proprio di m e del valore di α (contenuto in β). Considerare quindi l'effettivo range di scansione permette di accorpare le coppie di α e m simili senza avere il rischio di perdere informazioni e rendere vano il calcolo precedente.

Il concetto base è che si possono valutare assieme quelle soluzioni che hanno una m simile e come alpha prendere il valore massimo.

Il primo metodo di fitting scelto è stato quello di utilizzare una regressione polinomiale andando a interpolare le coppie di valori aventi m in ascissa e α come ordinata. Il primo step è stato quindi quello di ottenere una funzione polinomiale tramite il comando $poly = poly(m, \alpha)$. Successivamente si è proceduto con il calcolo dei massimi locali. Tramite script Matlab, usando un set di valori presi in maniera randomica dal data set ottenuto tramite lo script "DataSetManovellismo.m", si ottiene il seguente risultato visibile in figura 4.9.

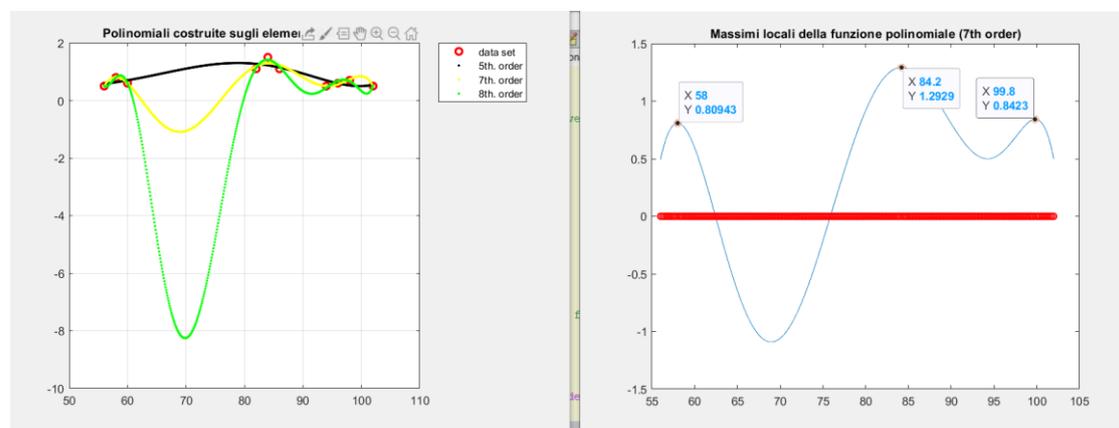


Figura 4.9: Regressione polinomiale con elementi randomici presi dal dataset

Tale metodo è stato successivamente scartato in quanto l'uso dell'interpolazione, per quanto la si ottimizzi, porta con sé degli errori che non sono facilmente recuperabili. Gli output, pur essendo nell'intorno desiderato, presentano dei valori di α al di sotto o al di sopra di quelli effettivi, figura 4.10, rendendo del tutto vane le operazioni di fitting.

```
matrix =
    56.0000    58.0000    60.0000    82.0000    84.0000    86.0000    94.0000    96.0000    98.0000
    0.5000    0.8000    0.6000    1.1000    1.5000    1.1000    0.5000    0.6000    0.7000

localMaxMatrix =
    58.1000    84.3000    99.9000
    0.8094    1.2929    0.8423
```

Figura 4.10: risultati della regressione polinomiale

Partendo sempre dalla stessa logica, ovvero la ricerca dei massimi locali, è stato implementato un secondo algoritmo di fitting all'interno della function SMF.m che è suddividibile, a sua volta, in tre sotto algoritmi interni come mostrato nel diagramma di figura 4.11.

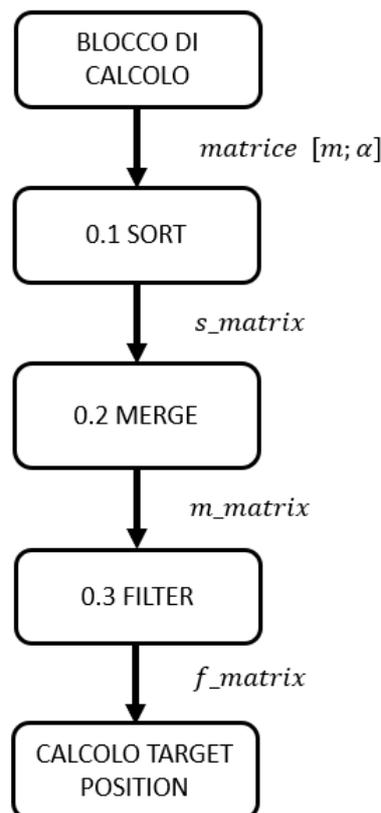


Figura 4.11: schema a blocchi dell'algoritmo di fitting SMF

In ingresso, l'algoritmo riceve una matrice la cui prima riga è costituita da valori di m mentre la seconda riga presenta i corrispettivi α calcolati tramite il modello del "falso manovellismo". Questa matrice entra quindi nel primo blocco che ordina i valori per m crescenti generando una seconda matrice s_matrix (sort matrix). A questo punto, si entra nel secondo blocco che tramite un merge si occupa di accoppiare tutte quelle

coppie di valori che rientrano all'interno del range di merge su m , dando in uscita la matrice m_matrix (merge matrix). Tale matrice presenterà degli zeri al posto dei valori scartati nel merge, quindi nel terzo blocco si effettua un vero e proprio filtraggio dei valori, in modo da ottenere la matrice f_matrix (fitted_matrix) con le correzioni finali da passare agli assi. In figura 4.12 viene riportato il funzionamento del blocco di merge.

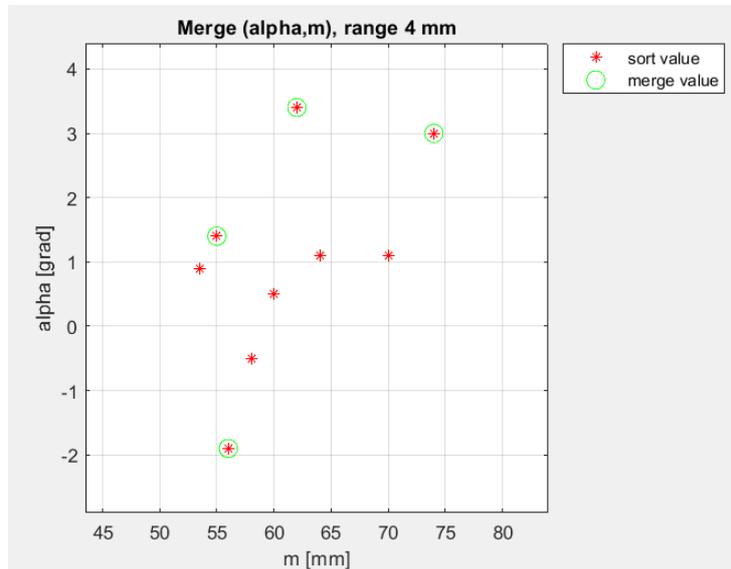


Figura 4.12: risultati ottenuti dalla function SMF

Con l'asterisco rosso vengono rappresentati tutti i valori, precedentemente ordinati in funzione di m dal blocchetto di sort. I valori cerchiati in verde corrispondono alle coppie "mergiate" mentre i restanti vengono scartati.

In figura 4.13. Viene invece riportato un esempio del funzionamento dell'algoritmo dando in ingresso una serie di valori presi sempre dal data set. Si può facilmente notare l'evoluzione che la matrice iniziale subisce ogni qual volta che attraversa uno dei blocchi di calcolo. L'algoritmo funziona, chiaramente, per valori positivi e negativi di α .

```

matrix =
  55.0000  70.0000  58.0000  62.0000  74.0000  56.0000  60.0000  64.0000  53.5000
  1.4000  1.1000  -0.5000  3.4000  3.0000  -1.9000  0.5000  1.1000  0.9000

s_matrix =
  53.5000  55.0000  56.0000  58.0000  60.0000  62.0000  64.0000  70.0000  74.0000
  0.9000  1.4000  -1.9000  -0.5000  0.5000  3.4000  1.1000  1.1000  3.0000

m_matrix =
  0  55.0000  0  56.0000  0  0  62.0000  0  74.0000
  0  1.4000  0  -1.9000  0  0  3.4000  0  3.0000

f_matrix =
  55.0000  56.0000  62.0000  74.0000
  1.4000  -1.9000  3.4000  3.0000

```

Figura 4.13: evoluzioni delle matrici intermedie durante l'algoritmo SMF

4.5 Scostamenti massimi dal valore nominale causati dal fitting e valutazione del range di merge

Per ottenere le migliori prestazioni dal sistema e validare l'algoritmo di fitting bisogna verificare quali scostamenti vengono introdotti, a valle del fitting stesso, rispetto alle coppie nominali di α e m in ingresso. Valutato ciò, sarà poi possibile stimare un opportuno range per effettuare il merge.

Siano α_f e m_f una generica coppia di valori di alpha e m in uscita dall'algoritmo di fitting, i punti che faranno riferimento a tali valori rientrano in quattro casi:

1. $\alpha < \alpha_f$ e $m < m_f$
2. $\alpha < \alpha_f$ e $m > m_f$
3. $\alpha \cong \alpha_f$ e $m < m_f$
4. $\alpha \cong \alpha_f$ e $m > m_f$

Nel caso (1) il punto richiede uno spostamento del montante minore di quello preso dal fitting ma presenta una distanza dal centro di rotazione minore, viceversa, nella condizione (2) il raggio di manovella associato al punto è maggiore di quello in uscita dal fitting. Gli altri due casi si hanno quando i valori di alpha del punto da mergare sono uguali o prossimi a quelli di alpha fitting. Come vedremo questi due ultimi casi sono riconducibili ai primi.

4.5.1 Primo caso: rotazione montante $\alpha < \alpha_f$ e raggio di manovella $m < m_f$

Come visto in precedenza, ad α_f ed m_f corrisponde un valore di x_{manov_f} e questo sarà il parametro passato all'asse di traslazione della tavola per correggere la posizione. Se si calcola adesso lo spostamento del piede di biella x_{manov} a parità di α_f ma in funzione del raggio di manovella effettivo del punto avremo che

$$x_{manov} > x_{manov_f}$$

Ovvero, a parità di angolo di biella, per un raggio di manovella minore si avrà uno spostamento del piede di biella richiesto maggiore. Graficamente è facile vedere che, avendo passato uno spostamento di biella minore di quello richiesto dal modello di manovellismo, il tutto si traduce con l'avere il centro del trapezio di scansione al di fuori del cerchio che rappresenta la traiettoria che il punto C, di raggio m, genera durante la rotazione della tavola.

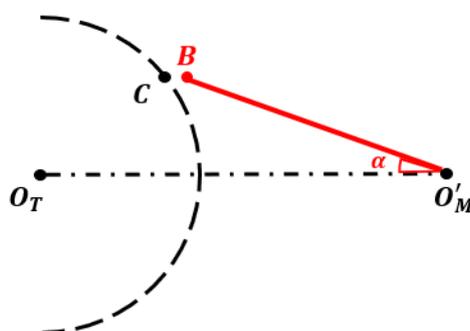


Figura 4.14: Rappresentazione grafica del caso $\alpha < \alpha_f$ e raggio di manovella $m < m_f$

In figura 4.15, viene proposto uno zoom dell'immagine precedente in cui il punto B corrisponde al centro del trapezio di scansione, mentre il punto C corrisponde al punto della superficie del pezzo sul quale è stato elaborato il vettore uscente e di cui prendiamo il valore di m . Si vede come il punto C non corrisponde più al centro del trapezio ("distacco" biella - manovella) e che quindi non verrà acquisito in quella posizione ma sarà intercettato dal fascio laser in nuova posizione detta C'.

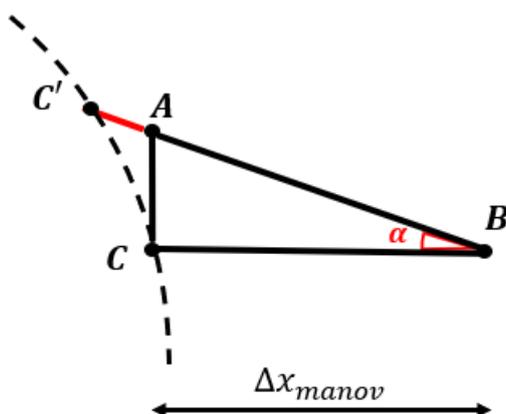


Figura 4.15: particolare della figura precedente

La distanza che interessa calcolare è il tratto C'B dato dalla somma di $AB+AC'$. Essa, infatti, deve essere minore della distanza che sussiste tra il centro ed i margini inferiori e superiori del trapezio di scansione. Se ciò non accadesse il punto C' sarebbe al di fuori del range di scansione rendendo vano il procedimento di calcolo traiettorie e di fitting. Partendo dai punti C e B si può costruire il triangolo rettangolo ABC avente come cateto CB la distanza che manca a C per sovrapporsi al trapezio di scansione. Tale valore è dato quindi da:

$$CB = \Delta x_{manov} = x_{manov} - x_{manov_f}$$

Essendo poi AB il proseguo del fascio laser, salirà con una pendenza nel piano pari proprio ad α_f .

Il tratto AB può essere calcolato come segue:

$$AB = \frac{CB}{\cos \alpha_f}$$

Mentre il tratto AC:

$$AC = AB * \sin \alpha_f$$

A questo punto è utile effettuare un'analisi delle grandezze degli elementi in gioco. Come è visibile dagli andamenti di α e x_{manov} proposti nel paragrafo precedente, il maggior valore di α trovato, che corrisponde alla peggior condizione possibile, è pari a circa 9° . Il sistema però presenta degli spostamenti elevati del piede di biella per raggi di manovella superiori ai 90mm. Per semplificare la discussione si ipotizzi che il sistema possa essere utilizzato fino a pezzi aventi raggio medio pari proprio a 90mm. In questa condizione l' α massimo trovato è pari a 6° .

Prendendo come riferimento tale valore di alpha si ha che:

$$AB = \frac{CB}{\cos(6^\circ)} \cong CB = \Delta x_{manov}$$

Avendo ipotizzato di essere nell'intorno di $m = 90\text{mm}$ ed essendo nel caso $m < m_f$, si ipotizzi di avere un raggio di manovella associato al punto C pari a $m = 88\text{mm}$ (che implicitamente equivale a prendere un range di fitting pari a 2 mm). Dalle solite formule del manovellismo si ottiene:

$$x_{manov_{88}} = 24.40 \text{ mm}$$

$$x_{manov_{90}} = 23.80 \text{ mm}$$

$$CB = \Delta x_{manov} = x_{manov_{88}} - x_{manov_{90}} = 0.6 \text{ mm}$$

Mentre

$$AC = AB * \sin \alpha_f \cong AB * 10^{-1} = \Delta x_{manov} * 10^{-1} = 0.06 \text{ mm}$$

L'incremento di ordinata è di un ordine di grandezza inferiore rispetto al delta di spostamento di biella.

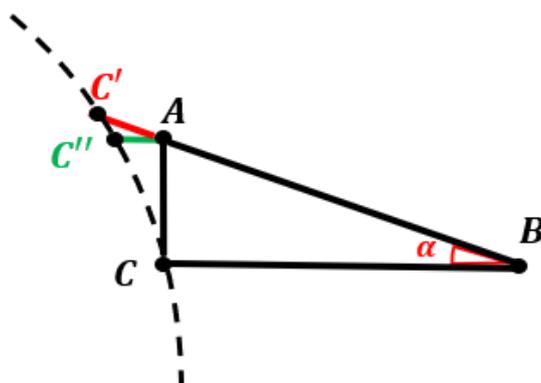


Figura 4.16: costruzione grafica A

Non resta quindi che calcolare l'ultimo tratto di interesse AC'. Se si considera il punto di costruzione C'', figura 4.16, dato dalla proiezione del punto A sulla circonferenza, esso risulta essere molto prossimo al punto C'. L'obiettivo si traduce quindi nel trovare il segmento AC'.

Il tratto AC non è altro che l'incremento in seno dato dalla rotazione di C su C'' ovvero è la differenza tra i segmenti C''K e CH di figura 4.17. Essendo noto β_C come l'angolo di manovella relativo al punto C (funzione di m_c e α_f), è possibile calcolare il nuovo angolo di manovella relativo al punto C''. Nell'esempio applicativo:

$$\beta_C = 40.79^\circ$$

$$\beta_{C''} = \sin^{-1} \left(\sin \beta_C + \frac{AC}{m} \right) = 40.84^\circ$$

Come si può vedere l'angolo risultante dal calcolo risulta essere maggiore del primo confermando l'intersezione del cerchio con il centro trapezio ad una quota maggiore.

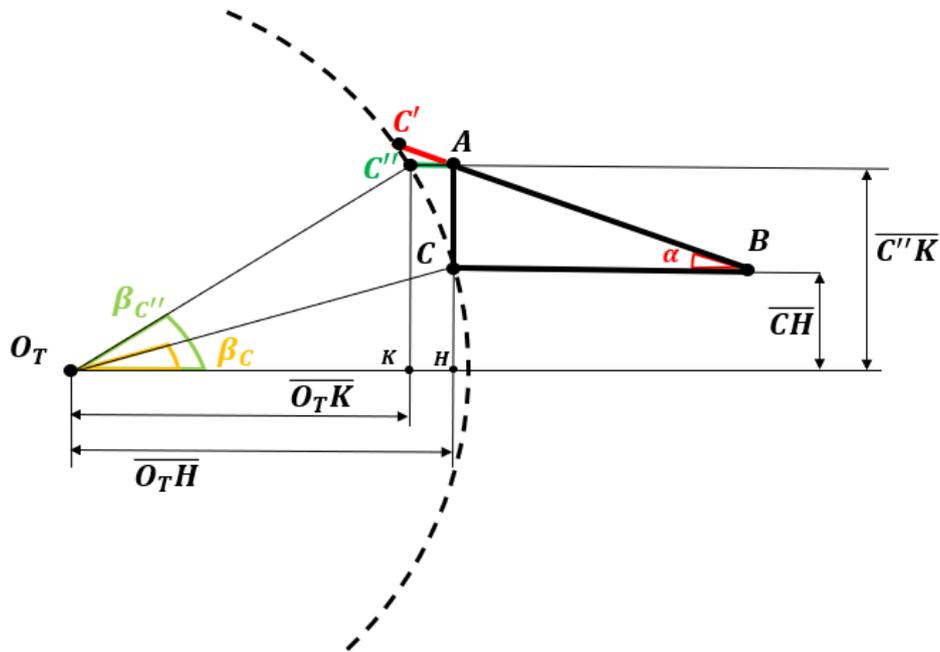


Figura 4.17: costruzione grafica per il calcolo di AC''

Noto $\beta_{C''}$, è possibile adesso calcolare la variazione in coseno tra i due angoli che sarà proprio il segmento AC'' :

$$O_TH = m * \cos \beta_C$$

$$O_TK = m * \cos \beta_{C''}$$

$$AC'' = m(\cos \beta_C - \cos \beta_{C''}) = 0.05 \text{ mm}$$

Assumendo $AC' \cong AC''$ per le solite considerazioni relative a $\cos(\alpha_f) = \cos(6^\circ) \cong 1$, si può quindi scrivere che:

$$BC' = AB + AC' \cong BC + AC'' = 0.65 \text{ mm}$$

Tale valore è ampiamente coperto dal range di scansione del trapezio. Ne risulta che il range di fitting ipotizzato pari a 2mm sia molto conservativo se si considera un raggio di manovella massimo nell'intorno di 90mm.

Nel caso in cui il punto presenti un valore di $\alpha \cong \alpha_f$ non si avrebbe nessuna perdita di dati in quanto esso verrebbe intercettato ad una quota maggiore che garantisce quindi l'abbattimento dell'angolo theta.

4.5.2 Secondo caso: rotazione montante $\alpha < \alpha_f$ e raggio di manovella $m > m_f$

Da un punto di vista metodologico, si procede come per il caso precedente. La differenza è che, essendo il raggio di manovella del punto C (m) maggiore di quello usato per il calcolo (m_f), la correzione sullo spostamento effettivamente passato alla tavola è maggiore di quella richiesto:

$$x_{manov} < x_{manov_f}$$

La tavola, quindi, è spostata in avanti ed il centro del trapezio B si troverà idealmente all'interno del cerchio di manovella, come visibile in figura 4.18.

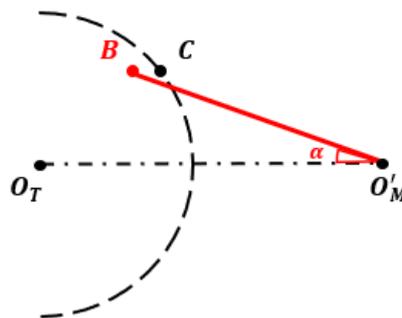


Figura 4.18: caso rotazione montante $\alpha < \alpha_f$ e raggio di manovella $m > m_f$

Ciò si traduce nel fatto che il punto C verrà intercettato prima dal raggio laser, nel punto C', e si potrebbe verificare il caso in cui la rotazione beta associata al nuovo punto C' non sia sufficientemente alta per abbattere il ϑ e rientrate quindi al di sotto del valore limite. In figura 4.19 viene proposto uno zoom per meglio capire la nuova posizione relativa tra centro trapezio e punto da acquisire.

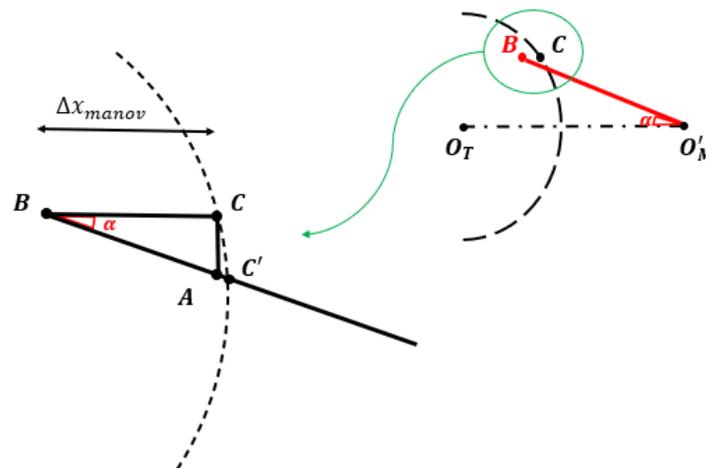


Figura 4.19: zoom figura precedente

Il rischio si intensifica nel caso in cui il punto B presenti un angolo di biella richiesto per abbattere ϑ molto vicino a quello di fitting:

$$\alpha \cong \alpha_f$$

Dato che lo scostamento sui beta è relativamente piccolo, la soluzione più adatta per ovviare a questa problematica è quella di essere maggiormente conservativi sul ϑ'_{lim} necessario al calcolo dei valori pre-fitting.

In conclusione, dall'analisi del fitting si ricavano le seguenti considerazioni sul sistema:

- Il range di lavoro ottimale si ha per un raggio di manovella compreso tra 50 e 90 mm;
- Il ϑ'_{lim} deve essere minore di 90° per evitare perdite di dati a valle del fitting (es: 88°);
- Per un ϑ massimo di 135° , l' α massimo sarà circa pari a 6° (massima movimentazione).

Nelle condizioni di massima movimentazione, tramite script Matlab, si è visto come imponendo un range di fitting di 4mm si ottengano compensazioni richieste lungo il raggio laser di +/-1.5mm. Nel caso di un profilometro LMI 2430A si ha un range di misurazione di 80mm, ovvero 40 mm di compensazione in entrambi i versi. Ciò consente l'utilizzo di un range di fitting relativamente ampio senza discostarsi troppo dal centro del trapezio, zona che permette di avere la miglior scansione possibile in termini di precisione e disturbi.

4.6 Gestione della correzione ed effetti nelle target position

Il grado di libertà posseduto dal montante riguarda la sola rotazione rispetto l'asse Z, ne consegue che la correzione lungo X, data dalla quantità x_{manov} , non potrà essere gestita da quest'ultimo. Si è scelto di tenerne conto durante la traslazione della tavola rotante (unica movimentazione lungo la direzione X), in quanto in virtù del moto relativo uno spostamento del montante corrisponde al medesimo spostamento della tavola ma in direzione opposta. Durante la fase di calcolo delle target position, che viene effettuata al termine della function di secondo livello "SMF.m", si procede con la correzione sulla posizione tramite la seguente formula:

$$x'_{tavola} = d_2 - x_{laser\&staffa} - m + x_{manov}$$

Dove allo spostamento della tavola viene sommata la quota parte di corsa aggiuntiva data dal modello del falso manovellismo (il centro tavola deve avvicinarsi al montante).

Questa formulazione a sua volta deriva dall'equazione base utilizzata all'interno della function di primo livello "MMR.m" che prevede una rotazione nulla del montante:

$$x_{tavola} = d_2 - x_{laser\&staffa} - m$$

Dove:

x_{tavola} , posizione del centro della tavola rotante;

d_2 , offset fisso (vedere tabella parametri di DH);

$x_{laser\&staffa}$, distanza tra il centro trapezio di scansione ed il centro nel montante che equivale alle dimensioni della biella del modello manovellismo;

m , è la lunghezza della manovella ovvero la coordinata x del punto con il vettore fuori range;

x_{manov} , spostamento del piede di biella che nel moto relativo diventa un incremento della X tavola dato che il montante è fisso.



Figura 4.20: rappresentazione grafica del calcolo della posizione target “traslazione tavola”

L’attivazione della function di secondo livello provoca delle modifiche anche sul file di testo finale “TargetPosition.txt”. In figura 4.21 vengono mostrati due file generati a partire da uno stesso modello CAD, opportunamente scelto, e con lo stesso numero di piani di scansione. Nel file di sinistra l’elaborazione è stata effettuata ponendo un angolo $\vartheta'_{lim} = 89^\circ$ invece in quella di destra si è scelto un angolo limite di 88° maggiormente conservativo.

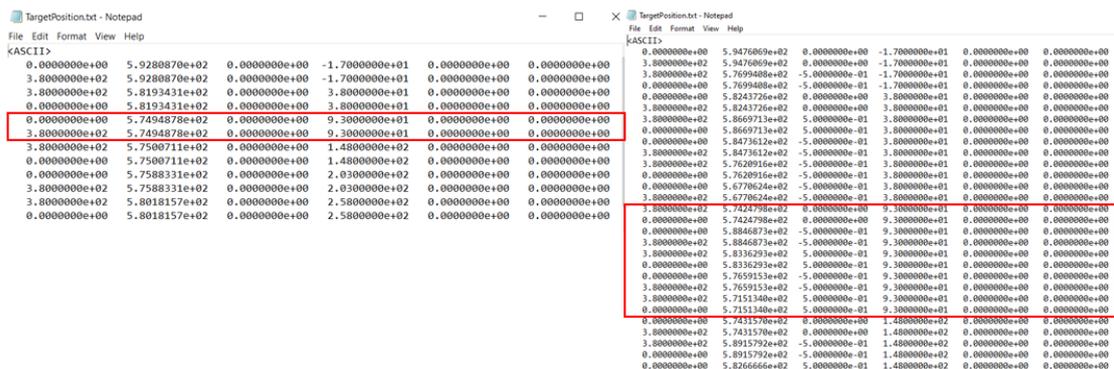


Figura 4.21: risultati dell’applicazione della SMF.m sul file TargetPosition.txt

In rosso è evidenziato il terzo piano di scansione alla quota di riferimento di 93mm. Nel primo caso ($\vartheta'_{lim} = 89^\circ$) è presente un’unica acquisizione suddivisa in una movimentazione A per il posizionamento tool ed una movimentazione B che prevede la sola rotazione della tavola. Con questo valore limite risulta evidente che la function di secondo livello non sia stata attivata in quanto non sono stati trovati vettori con angoli superiori a quello limite. Nel secondo caso invece ($\vartheta'_{lim} = 88^\circ$), allo stesso piano visto in precedenza, sono associate altre quattro acquisizioni con angoli del montante di -0.5° e 0.5° . In questo caso applicativo il range di fitting usato è stato posto a 4mm in modo da evidenziare il funzionamento della function “SMF.m”. Inserendo un valore più alto si riuscirebbe ad intensificare l’azione di merge e ridurre il numero di movimentazioni. Si sottolinea come il valore della seconda colonna, relativo alla target position da passare al secondo giunto che si occupa della traslazione tavola, decresca man mano che si aggiungono movimentazioni. Ciò si spiega dal fatto che nel calcolo della x_{tavola} rientra il valore di m del punto da acquisire con segno negativo e che tali valori sono ordinati dall’algoritmo di sort in ordine crescente.

5. Conclusioni e sviluppi futuri

Durante il progetto di tesi si è reso necessario validare tutte le varie fasi di sviluppo, presentate nei capitoli precedenti, tramite una serie di test specifici in modo da poter integrare al meglio le varie task, che portate avanti in parallelo, presentavano interazioni e punti in comune.

5.1 Test script SA-ML

Al fine di verificare la corretta importazione nell'ambiente di lavoro del file relativo al modello matematico da scansione, è stato utilizzato un generico pezzo avionico (splice di un B727) al momento disponibile nel reparto di ricerca e sviluppo. Sempre tramite quest'ultimo, è stato poi possibile validare lo script di SA che consente la generazione dei vettori normali alla superficie del pezzo lungo i vari piani di scansioni. Durante i test sono state evidenziate alcune criticità sulla generazione dei vettori in prossimità di fori ed asole. A seconda, infatti, della "qualità" del disegno CAD alcune superfici del pezzo possono presentarsi con normale riversata, causando quindi la generazione di vettori uscenti non utilizzabili. Per ovviare a questo problema sono stati inseriti dei filtri all'interno dello script Matlab che permettono di filtrare eventuali vettori con angoli di incidenza non adatti alla scansione.

5.2 Test protocollo di comunicazione SA-Matlab

Successivamente è stato testato il protocollo di scambio dati tra i due SW di calcolo, ovvero SpatialAnalyzer e Matlab. La necessità di dover lavorare sulla scrittura e cancellazione di file di "appoggio" ha influito sulla scelta di creare una directory specifica "WorkFolder_ARS" all'interno della quale inserire tutti i file di sistema, organizzati nelle rispettive sottocartelle. Durante questa fase non si sono evidenziate particolari criticità.

5.3 Test modello cinematico implementato su AVRIS

Definita e testata la fase di calcolo del processo di scansione, si è poi passati alla validazione del modello cinematico del sistema che è stato inserito all'interno del SW di controllo robotico AVRIS. Tale fase di test presentava due obiettivi diversi:

- Verificare la corretta posizione dell'end-effector e quindi il funzionamento della cinematica diretta;
- Verifica della corretta importazione delle nuvole di scansione e quindi della cinematica inversa.

Per la prima verifica è stato utilizzato un dataset costituito da una serie di posizioni target per ogni asse della struttura. Tale set di posizioni è stato poi passato manualmente ad ogni asse in modo da poter svolgere la fase di test senza dover passare dal codice PLC. Tale procedura è stata resa possibile in virtù del fatto che su TC3 è possibile configurare, e quindi utilizzare, un asse virtuale software collegato a quello reale. In figura 5.1 vengono infatti mostrati a sinistra i comandi di JOG (slow e fast), che permettono di pilotare il generico asse in velocità, mentre a destra viene mostrata

l'interfaccia che consente l'inserimento della posizione target da raggiungere e della velocità con la quale raggiungerla.

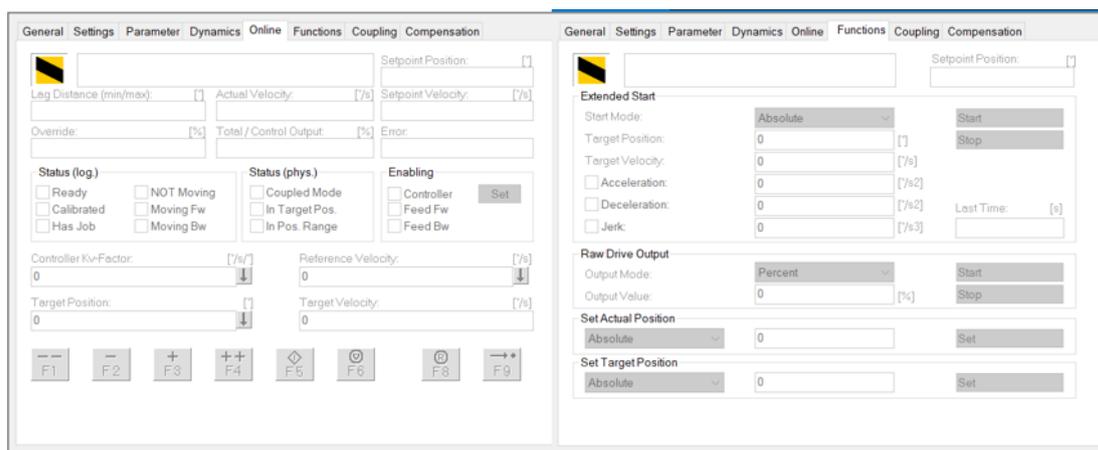


Figura 5.1: NC-axes, Online view (sx), Function view (dx)

Svolta quindi la movimentazione verso la posizione target sono stati confrontate le posizioni degli encoder assi e la posizione nello spazio dell'ultima frame coincidente con il centro del trapezio di scansione (vedere interfaccia AVRIS, figura 2.15 pag. 41).

Per il test sulla cinematica inversa è stato utilizzato un artefatto (soffietto di materiale plastico) al momento reperibile in reparto, caratterizzato da una geometria assialsimmetrica e avente delle facce a 45 gradi molto simili ad una eventuale filettatura, figura 5.2.



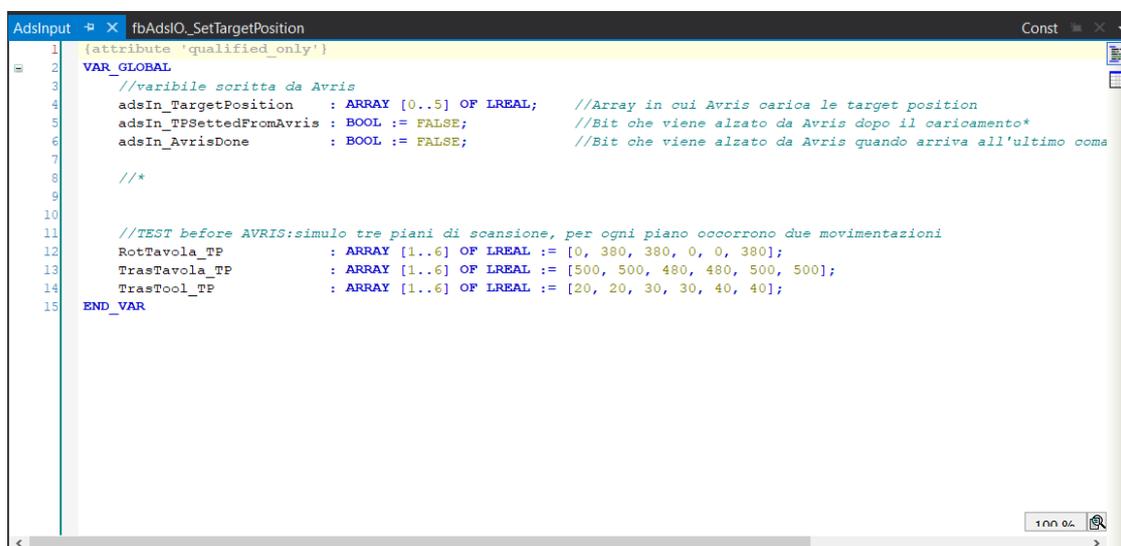
Figura 5.2: soffietto utilizzato per la fase di testing

Durante questa fase ci si è concentrati sul verificare che le nuvole di scansione fossero correttamente importate nello spazio rispetto al sistema di riferimento base, senza prestare particolare attenzione alla ricostruzione effettiva del componente scansionato. Ciò è dovuto al fatto che all'interno del modello cinematico erano ancora presenti i parametri di DH nominali presi dai disegni tecnici della documentazione e da rilevazioni manuali.

5.4 Test ciclo macchina nel PLC

Data la complessità del codice di controllo si è scelto di testarlo a blocchi in modo da riuscire a ridurre i tempi di debug e verificare al meglio l'effettivo funzionamento delle varie Function e delle azioni/metodi ad esse associati. Di seguito vengono elencate le varie fasi del test:

- Nella fase 1 dei test ci si è concentrati sulla macchina a stati per la gestione della movimentazione congiunta dei vari assi. Per farlo è stato usato un buffer statico relativo alle target position, implementato all'interno del codice e successivamente commentato, figura 5.3. L'obiettivo principale è stato quello di verificare la corretta esecuzione delle singole movimentazioni, in particolare che gli assi di posizionamento end-effector si attivassero alla fine di ogni rotazione della tavole tavola (ovvero alla fine della scansione) e mai prima;



```
1 {attribute 'qualified_only'}
2 VAR_GLOBAL
3 //variabile scritta da Avris
4 adsIn_TargetPosition : ARRAY [0..5] OF LREAL; //Array in cui Avris carica le target position
5 adsIn_TPSettedFromAvris : BOOL := FALSE; //Bit che viene alzato da Avris dopo il caricamento*
6 adsIn_AvrisDone : BOOL := FALSE; //Bit che viene alzato da Avris quando arriva all'ultimo coma
7
8 /**
9
10
11 //TEST before AVRIS:simulo tre piani di scansione, per ogni piano occorrono due movimentazioni
12 RotTavola_TP : ARRAY [1..6] OF LREAL := [0, 380, 380, 0, 0, 380];
13 TrasTavola_TP : ARRAY [1..6] OF LREAL := [500, 500, 480, 480, 500, 500];
14 TrasTool_TP : ARRAY [1..6] OF LREAL := [20, 20, 30, 30, 40, 40];
15 END_VAR
```

Figura 5.3: buffer statico delle target position usato in fase di testing

- Nella seconda fase, confermato il corretto funzionamento del controllo movimentazione assi, è stata testata la comunicazione ADS tra PC-Server e PC-client per il trasferimento delle posizioni target dall'albero di AVRIS verso il buffer dinamico visibile in alto sempre in figura 5.3;
- Nella terza fase sono stati testati la fase di homing (che consente di portare gli assi nelle rispettive posizioni di zero), la movimentazione di disimpegno e uscita (per un eventuale cambio pezzo) ed i comandi di reset e stop visibili nell'interfaccia di figura 2.22 a pag. 51;
- Nella quarta ed ultima fase sono state testate le variabili interne per la gestione del trigger HW verso i drive Kollmorgen ed il profilometro laser, più le variabili in scrittura ADS per la gestione dell'esportazione della nuvola di punti e dei vari piani di scansione.

Al termine dei vari test, dopo aver effettuato le varie modiche e correzioni, è stato possibile calcolare il tempo che intercorre tra l'acquisizione di una lama di scansione del profilometro laser e la successiva. Il valore medio di questo delta temporale risulta essere di circa 70 MS, ovvero di circa 14 scansioni al secondo. Tale valore risulta essere

in linea con i target aziendali se si considera che Axist con il Robot è riuscita a raggiungere un tempo medio di circa 20 MS tra due scansioni successive. Inoltre, con questo tempo di acquisizione ed una velocità angolare della tavola di $100^\circ/s$ si ha una lama di scansione ogni 7° circa.

5.5 Risultati sulla scansione a valle della calibrazione

Come già visto nel terzo capitolo, tramite un laser tracker Leica ed il SW ad esso associato, è stato possibile effettuare una calibrazione del sistema. Tale procedura ha consentito di effettuare una prima correzione geometrica ai parametri di DH nominali. Inserendo quindi i nuovi valori trovati all'interno della cinematica di AVRIS, si è andati a migliorare la cinematica inversa del sistema e ottenere una ricostruzione, rispetto al sistema di riferimento base, nettamente più precisa. In figura 5.4 viene proposto il confronto tra due tipi di acquisizione tramite un'immagine presa dal file di lavoro SA. Il sistema è visto dall'alto e si possono infatti distinguere il profilometro laser sulla destra ed il montante verticale mentre al centro sono presenti in rosso il carrello che consente la traslazione della tavola ed in nero il piatto della tavola rotante.

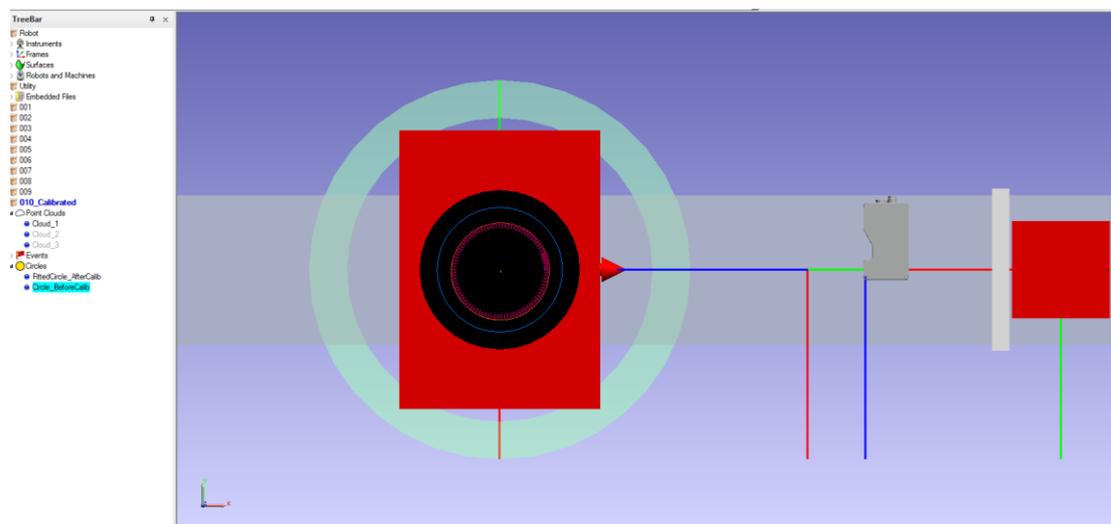


Figura 5.4: confronto acquisizione prima e dopo la calibrazione

Il cerchio blu è stato elaborato partendo da una scansione con il sistema non calibrato e rappresenta la circonferenza del cilindro (vista dall'alto), ottenuto tramite un fitting sulle varie cloud di scansione (cloud_1, cloud_2, cloud_3 visibili nell'albero di SA a sinistra) riferite al soffietto di figura 5.2.

Più internamente si trovano in rosso la cloud riferita ad una scansione con i parametri calibrati ed in arancione la circonferenza del cilindro fittato. Come è evidente dalla figura, la circonferenza calibrata risulta avere un diametro inferiore rispetto alla circonferenza calcolata sulle nuvole del sistema non calibrato. In particolare, a fronte di un diametro nominale del componente, misurato tramite calibro, di 200mm si è passati da una circonferenza ricostruita di diametro 250mm cm ad una avente diametro 195.72 mm come visibile in figura 5.5 dove viene riportata la tabella di SA con le informazioni relative alle due circonferenze.

010::Geometry List - Circles (mm)									
Reported in Robot::Base									
Name	X	Y	Z	I	J	K	Radius	Diameter	
FittedCircle_AfterCalib	1.09	-3.62	465.96	0.03	0.00	1.00	97.86	195.72	
Circle_BeforeCalib	0.00	0.00	460.00	0.00	0.00	1.00	125.00	250.00	

Figura 5.5: caratteristiche geometriche prima e dopo la calibrazione

Si nota come il centro della circonferenza calibrata non sia coincidente con l'origine degli assi X e Y. Ciò è dovuto al fatto che, non essendoci un sistema di centraggio, tra prima e seconda acquisizione si è leggermente decentrato il pezzo rispetto al centro tavola. La differenza di circa 4 mm tra il diametro nominale e quello misurato è dovuta ad una non linearità dell'asse di traslazione tavola e di traslazione montante.

5.6 Sviluppi futuri

Dal punto di vista dell'architettura Software, il sistema può considerarsi ultimato. La comunicazione EtherCAT – ADS, socket PTP ed il protocollo di comunicazione SAML consentono di gestire in piena autonomia il trasferimento dei dati e delle variabili di controllo, mentre la suite AVRIS consente di configurare il sistema di misura ed eventualmente gestire ulteriori gradi di libertà assi rotoidali che prismatici.

Per quanto riguarda invece il codice PLC, un ramo di sviluppo potrebbe comprendere la funzione e la procedura di latch degli encoder assi e la macchina a stati che consente l'aggiornamento delle variabili ADS del ciclo trigger. Al momento, infatti, l'unico latch HW viene effettuato sull'encoder della tavola rotante che presenta un modulo drive Beckhoff. In questo caso, il segnale di trigger in uscita dal modulo Beckhoff di digital output viene utilizzato come input nel modulo drive per effettuare il congelamento delle posizioni. Tale procedura consente di avere un congelamento rapido e preciso del valore dell'encoder tavola ad ogni lama di scansione. Per gli altri assi non è stato possibile implementare un trigger HW in quanto l'azionamento risulta essere costituito da Drive Kollmorgen e si è virati quindi su un trigger di tipo SW per il congelamento delle posizioni. Tale procedura, dovendo attenersi al tempo ciclo PLC, ha un effetto peggiorativo sul tempo che intercorre tra due acquisizioni. Le possibili soluzioni sono le seguenti:

- Configurare i drive a ricevere un segnale digitale per il congelamento della posizione tramite il software Kollmorgen WorkBench;
- Sostituire i drive Kollmorgen con drive Beckhoff compatibili ed avere un integrazione completa dell'intero sistema.

La prima soluzione, seppur più economica, non garantirebbe di avere la migliore prestazione dal sistema che si avrebbe, invece, sfruttando un azionamento interamente Beckhoff.

Per migliorare il tempo ciclo di acquisizione, ovvero da quanto nasce il segnale trigger a quando ad AVRIS arrivano i bit di avvenuta esecuzione dal PLC e dal profilometro, le possibili migliorie sono:

- Sfruttare l'attuale funzione e farla girare in una sotto task con velocità maggiore (es. 5MS);
- Utilizzare altre funzioni della libreria;

- Utilizzare dei moduli Beckhoff di fast output e fast in.

Allo stato attuale il set di posizioni da passare agli assi viene calcolato in maniera semiautonomo da SA e Matlab. Una possibile alternativa è costituita dall'implementazione della modalità MAPS "Manual Axes Position Scanning" tramite la quale si porta manualmente il tool nelle posizioni desiderate per poi salvare i corrispondenti valori degli assi (latch esterno manuale). Una volta definito il piano di scansione, tramite un apposito comando da interfaccia utente si termina la fase di pianificazione della movimentazione ed inizia la scansione del pezzo.

I vantaggi di avere la possibilità di pianificare in manuale sono :

- Possibilità di definire un piano di scansione personalizzato e mirato ad aree specifiche del pezzo;
- Possibilità di aggiungere nuove scansioni a quelle calcolato in modo automatico;
- Maggiore flessibilità del sistema rispetto a pezzi con particolari geometrie.

Da terminare ci sono la gestione degli allarmi e l'interfaccia uomo macchina che al momento è stata sostituita dalla funzione "Visu" interna al PLC, paragrafo 2.8, che però è accessibile solo dagli sviluppatori e non da potenziali clienti. Un esempio di HMI è fornita da quella usata per il Robot AVRIS, figura 5.6.

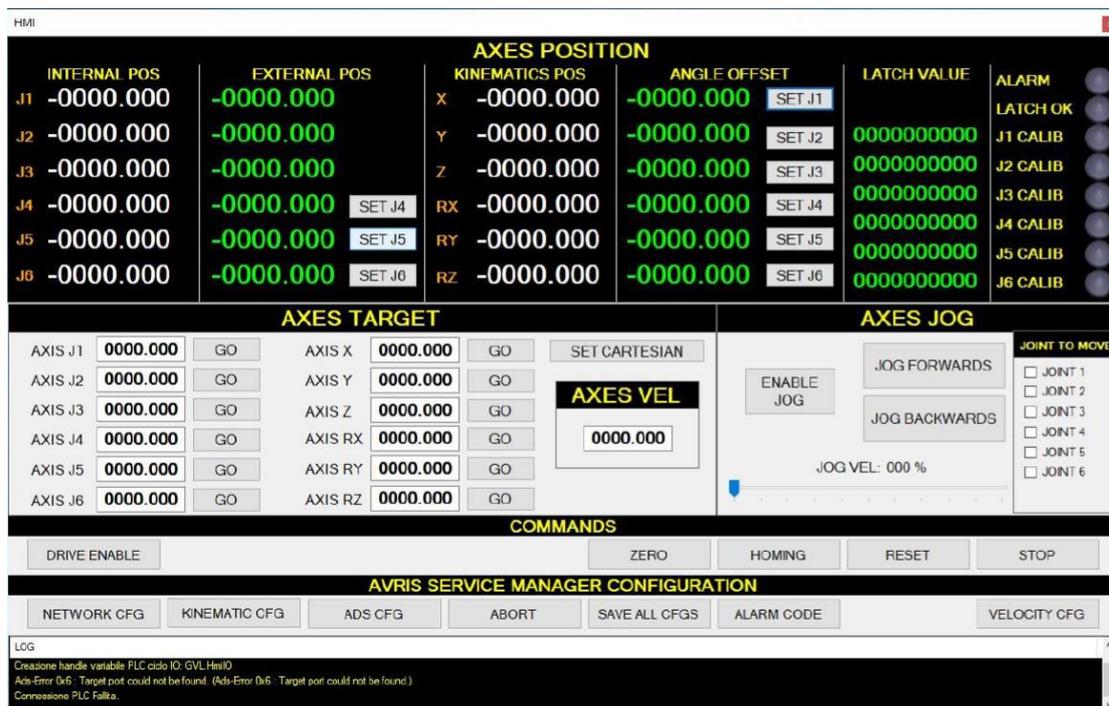


Figura 5.6: HMI del Robot AVRIS

Partendo da questa bisognerebbe:

- semplificare l'area AXES POSITION, in quanto quattro giunti presentano solo encoder interni;
- Mantenere AXES TARGET in modo da visualizzare il set passato da Matlab;
- Implementare nell'area AXES JOG e COMMANDS la modalità MAPS.

Altro importante ramo di sviluppo è costituito dall'ideazione e costruzione dell'ultimo asse del sistema che governa la rotazione del profilometro laser. Le possibili alternative per movimentare il tool in fase di valutazione sono le seguenti:

- Servo-azionamento con leveraggio;
- Motore passo-passo;
- Motore brushless con sistema di riduzione (cinghia) ed albero.

Tutte le possibilità presentano sia vantaggi e che svantaggi. Nel caso del servomotore, alla economicità e facile reperibilità dei componenti si contrappone la difficoltà di progettazione del leveraggio e di tracciabilità della movimentazione. Per il motore passo-passo si hanno costi più alti ma la possibilità di non inserire alcun encoder esterno essendo di base molto preciso. Nella configurazione con motore-cinghia si avrebbero i costi della componentistica e degli encoder esterni, che potrebbero però essere "abbattuti" se si pensa di riutilizzare uno dei motori del secondo montante che è attualmente fermo.

Per terminare la calibrazione del sistema andrebbero tracciati i vari assi lineari della traslazione tavola e traslazione tool, così come individuare con maggiore precisione l'asse di rotazione montante per verificare eventuali cedimenti durante la movimentazione. A queste si accompagna anche l'esigenza di elaborare una procedura di calibrazione del tool in flangia. Infatti, dopo aver calibrato il sistema bisogna essere certi di come sia montato il profilometro in flangia in modo da determinare con certezza la posizione nello spazio del trapezio di scansione. Una possibile soluzione sarebbe quella di usare un tool di calibrazione costituito da un cilindro di diametro certificato ed acquisirlo in diverse configurazioni del montante e della flangia tool. Dalla ricostruzione del cilindro si estrapolano poi gli offset di correzione da applicare in flangia.

6. Bibliografia

- Axist S.r.l. (2020). *Axist S.r.l.* Tratto da <https://www.axist.it/>
- Beckhoff. (2020). *BECKHOFF*. Tratto da <https://infosys.beckhoff.com/>
- Faro. (2020). *Faro*. Tratto da <https://www.faro.com>
- GUDEL. (2003). Documentazione sistema.
- LMI. (2020). *LMI*. Tratto da <https://lmi3d.com/>
- NEWPORT. (2018). Data Sheet High-Performance Precision Rotation Stages.
- Spatial Analyzer. (2020). *Spatial Analyzer*. Tratto da <https://www.kinematics.com/spatialanalyzer/>
- Zeiss. (2020). *Zeiss*. Tratto da <https://www.zeiss.it/metrologia/home.html>