UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA IN INGEGNERIA MECCATRONICA

TESI DI LAUREA MAGISTRALE

INDUSTRIAL ETHERNET & MOTION CONTROL: DUE BUS A CONFRONTO

Relatore: Ch.mo Prof. GIOVANNI BOSCHETTI

Laureando: SIMONE VALMORBIDA

Matricola 601370-IMC

ANNO ACCADEMICO 2011-2012

Dedico questa tesi ai miei genitori,
che con la loro pazienza e supporto
mi hanno sostenuto in questi studi.

Sommario

In questo elaborato vengono analizzate due reti di comunicazione industriali basate sul mezzo fisico di ethernet.
Le due reti in questione sono SERCOS III®di cui Bosch Rexroth detiene l'esclusiva sull'utilizzo e EtherNet/IP™utilizzata principalmente da Allen-Bradley nel mercato americano e Rockwell Automation per il mercato europeo.
Per ciascuna rete si analizzano le specifiche hardware necessarie al funzionamento, il protocollo di comunicazione adottato, ponendo particolare attenzione alla fase iniziale di parametrizzazione dei dispositivi.
Lo studio delle reti viene svolto utilizzando parallelamente sia il software proprietario del dispositivo, con il quale vengono programmate le movimentazioni, parametrizzata la rete e i dispositivi, sia il programma open source, Wireshark, con il quale si cattura il traffico che circola in rete al fine di leggere, interpretare e dare rappresentazione grafica dei pacchetti e dei dati contenuti.
Questa duplice analisi ha permesso di studiare la struttura di comunicazione della rete in modo indipendente dalla personalizzazione e veste grafica che Bosch Rexroth o Allen Bradley hanno voluto dare.
L'analisi viene conclusa testando le due reti con alcuni programmi orientati al motion control dove si analizza come vengono gestite le movimentazioni e le traiettorie di due slave.
I test condotti sulla rete SERCOS III ha evidenziato una struttura non standard ethernet con un complesso set di regole proprietarie, che permettono di controllare e sincronizzare ogni aspetto della comunicazione e della gestione del moto.
Sotto il profilo software invece è stata riscontrata una certa difficolta nell'utilizzo del programma, dovuta probabilmente a difetti legati alla particolare versione in uso.
Il medesimo lavoro su rete EtherNet/IP ha fatto apprezzare i vantaggi offerti dal protocollo ethernet non modificato di questa rete, che hanno permesso di utilizzare hardware generico per fare misure. Le prestazioni complessive registrate sia in rete che su applicazioni di Motion Control sono state inferiori rispetto a SERCOS. E la struttura della comunicazione, il protocollo CIP, si è dimostrato piu complicato da studiare.
L'ambiente di sviluppo dei programmi invece ha piacevolmente sorpreso, offrendo allo stesso tempo un potente editor per la programmazione e un'interfaccia semplice per la gestione dell'impianto.

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Introduzione

Le reti di comunicazione industriale costituiscono l'infrastruttura attraverso la quale i vari componenti di un processo produttivo scambiano le informazioni tra loro. Questa infrastruttura è formata da una parte fisica che costituisce l'hardware necessario a comunicare quali i cavi, le interfacce di trasmissione e ricezione dei dati, l'unità di controllo centrale, i dispositivi di gestione del traffico in linea, etc. A questa va aggiunta una parte software chiamata protocollo di comunicazione che determina, attraverso un set di regole come a livello logico le informazioni circolano sui dispositivi hardware.
Le reti industriali hanno vissuto una certa evoluzione negli anni, seguendo con ritardo le innovazioni introdotte nelle reti general purpose come vengono definite le normali reti da ufficio o domestiche. Questo ritardo è dovuto principalmente alla differenza di obiettivi che si pongono queste reti:

Reti industriali	Reti consumer
Modeste quantità di dati da trasferire (Byte o KByte)	Elevate quantità di dati da trasferire (MByte o GByte)
Massima importanza all'aspetto temporale e deterministico della comunicazione	Massima importanza alla quantità di dati trasferiti (throughput della rete)
Necessità di comunicazioni Real-Time con Jitter il piu ridotto possibile	Interesse alle performance medie della rete
Richiesta massima affidabilità della rete, con meccanismi ridondanti per garantire il funzionamento anche in caso di guasto	É tollerato il guasto della rete
Hardware particolare conforme alle specifiche del produttore	Hardware generico conforme a specifiche internazionali

Negli ultimi anni si è registrata da parte di molte reti industriali, una decisa spinta verso l'adozione del livello fisico di ethernet per le loro comunicazioni. Questo fatto ha portato aziende come Siemens, Bosch-Rexroth, Allen Bradley e molte altre a offrire una soluzione ethernet ai propri clienti in alternativa al classico e sperimentato fieldbus.

In questa tesi verranno studiati due approcci alle comunicazioni industriali basate su ethernet. Nello specifico la rete SERCOS III proprietaria di Bosch-Rexroth e EtherNet/IP utilizzata da Allen Bradley/Rockwell Automation.
L'analisi di queste reti sarà condotto su due banchi prova, ognuno caratterizzato da un controllore che opera da Master sulla rete e due driver di media potenza che fungono da Slave. Proprio sfruttando questa configurazione con due azionamenti è stato possibile concentrare lo studio sugli aspetti di Motion Control che offrono i due ambienti di lavoro e sulle prestazioni offerte.

La linea guida seguita per analizzare le due reti si può riassumere brevemente come segue.
Per prima cosa utilizzando il software di sniffing Wireshark si è catturato parte del traffico che circolava in rete durante l'esecuzione di programmi base presenti come tutorial. A partire da questi pacchetti si è analizzato la struttura di comunicazione della rete, la forma e composizione dei pacchetti, ed eventuali particolarità. Per meglio comprendere le dinamiche presenti nell'impianto

si è spostata l'attenzione sulle prime fasi di vita della rete, cioè la procedura di accensione e quelle immediatamente successive. In questa fase infatti avviene parte della parametrizzazione degli slave, viene decisa la velocità di comunicazione e la struttura dei pacchetti. Questo ha permesso di comprendere completamente la struttura del pacchetto e avere un quadro completo della rete.
Una volta raggiunto un discreto livello di conoscenza sulla rete, si è passati ad analizzare l'aspetto di Motion Control offerto dai due impianti, si sono quindi studiati e tradotti nei rispettivi ambienti di lavoro tre programmi di test: un controllo di movimento sincronizzato, una camma elettronica e un elettronic gear.
Durante questi test sono stati catturati i pacchetti circolanti in rete e attraverso il software Matlab sono state riportate le traiettorie percorse dagli azionamenti. Proprio grazie a questa indipendenza dai software proprietari è stato possibile confrontare equamente i risultati ottenuti.

1.1 Organizzazione della tesi

Nel capitolo 2 si presenta un breve resoconto storico dell'evoluzione delle reti industriali, partendo dai collegamenti punto punto fino ad arrivare alla tecnologia Wireless; vengono inoltre evidenziate le principali differenze rispetto alle reti consumer.

Nel capitolo 3 si inizia un'analisi approfondita della rete SERCOS, si ripercorrono brevemente le tappe storiche che hanno portato alla creazione di SERCOS III, si analizza la struttura hardware utilizzata per costruire la rete e gli standard IEEE rispettati. Viene inoltre studiato il funzionamento della rete sotto il profilo logico della comunicazione, partendo dalla sua parametrizzazione off-line, fino ad arrivare alla descrizione di particolari eventi come l'ingresso in rete di un nuovo dispositivo. Si fornisce una descrizione della struttura del generico pacchetto SERCOS III riservandosi di discutere il campo dati in una sezione apposita piu avanti nella tesi.
Si conclude la trattazione teorica di questa rete descrivendo il meccanismo di sincronizzazione dei dispositivi, i metodi di diagnostica adottati, e le performance raggiungibili usando questa tecnologia.

Nel capitolo 4 si introduce la rete EtherNet/IP, valutando i requisiti hardware e la logica di funzionamento. Particolare spazio viene dato all'analisi del protocollo CIP, con i recenti sviluppi in campo di sincronizzazione e Motion Control chiamati CIP Sync e CIP Motion. Viene successivamente analizzata la forma del pacchetto tralasciando il campo dati. Si termina riportando le performances raggiungibili con questa rete nei vari casi di funzionamento.

Il capitolo 5 è dedicato alla presentazione del software Wireshark, con cui è stato possibile analizzare i pacchetti e di fatto realizzare lo studio delle due reti

Nel capitolo 6 si riportano i risultati dello studio dei pacchetti circolanti in rete; si riportano le varie tipologie di pacchetto circolanti in rete al variare dei programmi di motion control utilizzati. Per ciascuna tipologia viene riportata la dissezione con il relativo significato dei byte contenuti.

Nel capitolo 7 si analizzano gli impianti utilizzati per lo studio delle due reti, l'impianto Bosch Rexroth per la rete SERCOS III e l'impianto prodotto da Allen Bradley per la rete EtherNet/IP, vengono descritti i componenti hardware utilizzati quali il controllore, gli azionamenti e i motori. Viene inoltre dedicata un breve analisi anche ai rispettivi software di programmazione Indraworks e RSLogix 5000 e la procedura di parametrizzazione/commissioning dell'impianto.

Nel capitolo 8 si riporta la progettazione, realizzazione e i risultati ottenuti dei test di Motion Control. Descrivendo finalmente come è strutturato il campo dati e come si evolve cambiando

programma.
Si conclude questo elaborato con delle considerazioni generali sui due impianti e vengono proposti alcuni sviluppi futuri a partire da questa tesi.

Capitolo 2

Reti di comunicazione industriale

In questo capitolo verranno ripercorsi brevemente le tappe evolutive delle reti di comunicazione industriale, per meglio comprendere quali sono i requisiti che devono soddisfare queste reti, e le varie soluzioni storicamente adottate per farlo.
Per prima cosa è bene comprendere che i tempi di aggiornamento delle tecnologie in campo industriale sono molto maggiori rispetto alla controparte commerciale. Questo avviene per molteplici cause:

- Affidabilità:
  Le tecnologie impiegate in ambito industriale devono essere collaudate e offrire un livello di sicurezza e tolleranza ai guasti molto superiore rispetto all'ambito commerciale. Quindi prima di essere impiegata una nuova tecnologia deve passare un certo tempo perchè venga considerata sicura.
- Costo del cambio:
  Per introdurre una nuova tecnologia spesso non basta cambiare un elemento dell'impianto, ma è necessario cambiare l'intero hardware che riguarda quell'aspetto, quindi la spesa per adottare questa innovazione è superiore al semplice costo del nuovo componente.
- Compatibilità:
  Spesso si ha a che fare con impianti già costruiti di cui si desidera ammodernare solo un aspetto, conservando quanto già presente che soddisfa i nuovi requisiti. In questo caso ci si scontra con le compatibilità fra sistemi proprietari, che spesso obbliga a soluzioni di compromesso tra l'aggiornare il sistema e il riutilizzare l'hardware già presente.

I motivi sopracitati portano a far si che solo negli impianti costruiti ex-novo vengono introdotte le novità presenti sul mercato, ma i nuovi impianti sono solo una minoranza rispetto a quanti potrebbero adottare e beneficiare delle innovazioni introdotte. Di conseguenza solo su questi nuovi impianti è possibile valutare i pro e contro e allo stesso tempo la scarsa diffusione genera disinformazione e sfiducia che rallentando ulteriormente il processo di innovazione.

La scelta della rete di comunicazione determina che tipo di informazioni è possibile far circolare in rete e la frequenza con cui nuove informazioni sono disponibili ai dispositivi connessi in rete. Si comprende subito che l'efficenza complessiva dell'impianto sia intimamente legata a questa scelta, per questa ragione negli anni sono state sviluppate varie soluzione per meglio soddisfare i vari bisogni industriali. Queste reti possono essere riassunte in 4 grandi tipologie:

- Anello 4-20 mA
- Bus di campo
- Ethernet Industriale
- Wireless

2.1 Comunicazione basata sull'anello di corrente

Questo semplice modello di comunicazione si basa sulla chiusura di un anello di corrente tra il controllore e ogni dispositivo che si intende connettere alla rete garantendo quindi una comunicazione indipendente tra master e ogni slave. Gli elementi hardware presenti in rete sono il PLC

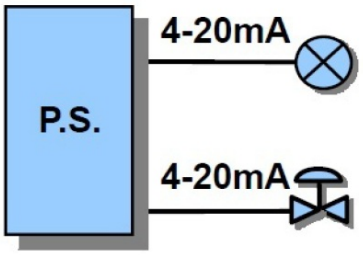


Fig. 2.1: Schematizzazione di un anello di corrente 4-20 mA

come controllore e un semplice cavo in doppino intrecciato e schermato per effettuare i collegamenti verso i dispositivi. La comunicazione utilizzava un generatore di corrente per generare i livelli logici, che a seconda del protocollo potevano portare informazioni analogiche o digitali: nello specifico, in digitale il livello logico 0 veniva associato a 4 mA di corrente circolante in rete, mentre i 20 mA venivano associati a 1.
Mentre in analogico venivano utilizzati anche i valori intermedi tra 4 e 20 mA per portare un'informazione più ricca.
Il vantaggio principale di questa struttura di comunicazione era che l'assenza di corrente o valori inferiori ai 4 mA sulla rete significava sempre e soltanto un guasto.
I principali difetti di questa rete erano legati al cablaggio della rete, per ogni dispositivo era necessario utilizzare un cavo. Questo fatto influiva negativamente sul costo, complessita di realizzazione e manutenzione dell'impianto, oltre a generare un limite insuperabile nel numero di dispositivi collegabili.

2.2 Bus di campo

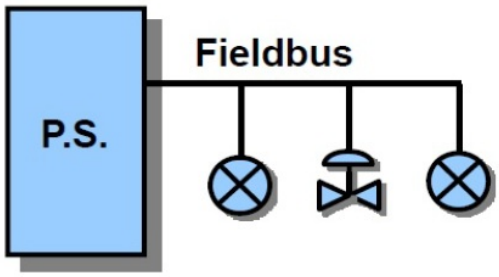


Fig. 2.2: Schematizzazione di un Bus di campo

Per superare i difetti dei collegamenti 4-20 mA in seguito al completamento dello standard ISA S50.02 si inizia a concepire un nuovo modo di connettere i dispositivi: a partire dal controllore centrale si crea un'unica dorsale (Bus) che percorre tutta la lunghezza dell'impianto, ai dispositivi che desiderano comunicare viene quindi richiesto solamente di connettersi a quest'ultima e attraverso di essa viene garantita la comunicazione con il resto dell'impianto. Nasce cosi il bus di campo (FieldBus).
Questo soluzione ha modificato radicalmente l'hardware dei dispositivi presenti sulla rete che, per gestire la comunicazione in modo strutturato, necessitano ora di un FPGA o ASIC che svolga le funzioni di interfacciamento tra dispositivo e bus.
Successivamente, sfruttando l'elettronica già presente a bordo degli slave, molti produttori hanno iniziato a fare delle elaborazioni locali dei dati raccolti sul campo prima di trasmetterli in rete, fornendo un nuovo modo di gestire la rete, decentrando parte del controllo dal master centrale

verso i dispositivi periferici.
Le altre novità introdotte dai Bus sono l'introduzione della comunicazione bidirezionale e l'aumento della quantita di dati che possono essere trasferiti in rete. La rete comincia ad essere usata oltre che per la gestione del processo produttivo, anche per la parametrizzazione e diagnostica dei dispositivi, quali lo stato degli slave, e altri dati utili per gestire la manutenzione dell'impianto.
Viene aumentata inoltre la velocità con cui circolano le informazioni in rete e, trattandosi di un mezzo di comunicazione condiviso, si inizia a porsi dei problemi sul determinismo della comunicazione e sulla necessità di avere le funzioni di Real-Time per gli allarmi.
Il determinismo garantisce che le informazioni prelevate dagli slave arrivino al master, e nello stesso modo i comandi del master raggiungano gli slave a istanti di tempo regolari chiamati comunemente tempo di ciclo. Creando cosi una sequenza di eventi sincronizzati tra loro entro un certa tolleranza definita dal jitter della rete determinabile a priori.
Contemporaneamente a questo tipo di comunicazione si richiede che un allarme segnalato da qualche parte dell'impianto viene registrato dal controllore entro un certo intervallo di tempo, in modo tale che possa reagire tempestivamente alla situazione di pericolo. La necessità di ottenere dei tempi di reazione certi da parte del impianto viene definita capacità di Real Time. Tutto questi miglioramenti hanno da un lato ridotto i costi delle connessioni tra i dispositivi, dall'altro hanno aumentato il costo dei dispositivi stessi a causa della maggior quantità di elettronica a bordo. Nel complesso questo ha portato il prezzo delle soluzioni basate su FieldBus ad essere decisamente maggiore rispetto agli anelli 4-20 mA, offrendo però tutta una serie di importanti benefici pratici.
Tutte queste idee sono state comprese e rielaborate da molte aziende a partire dagli anni '80 e '90 e hanno portato alla creazione dei famosi bus di campo Profibus (Siemens 1989) LonWorks, Inerbus, SafetyBUS etc. ognuno con specifiche e prestazioni differenti, in modo tale da rendere incompatibili tra loro la maggior parte dell'hardware.
A livello fisico i principali mezzi storicamente adottati sono RS232 e RS485 e la velocità di trasferimento dati raggiunta varia dai 9600bit/s ai 12Mbit/s ottenuti con Profibus. Il numero delle stazioni collegabili a ogni bus è di circa 20-30 mentre la lunghezza massima del bus è in funzione della velocità che si desidera raggiungere.
Attualmente le soluzione a Bus sono largamente presenti e utilizzate in azienda, in quanto garantiscono con la loro tecnologia consolidata e ampiamente testata una solida base per le comuni automazioni.

2.3 Ethernet industriale

La crescente complessita dei dispositivi da inserire nei processi produttivi, la sempre maggior spinta a decentrare negli slave parte dell'intelligenza presente nella rete, ha provocato un aumento della banda trasmissiva richiesta, cosa che i normali bus di campo non erano in grado fornire.
Questo ha portato i produttori di reti industriali a ricercare un nuovo mezzo trasmissivo per soddisfare le nuove esigenze. Questa ricerca ha vagliato molti standard e alla fine si è concentrata su ethernet in particolare sul suo protocollo a livello data link.
Ethernet offriva il vantaggio di essere gia ampiamente diffuso in ambito consumer, dove si era gia imposto come standard mondiale di fatto. Questo ha permesso di poter contare su un gran numero di produttori in grado di fornire dispositivi con prestazioni sempre più elevate a prezzi competitivi.

Ma l'ethernet general purpose non era adatto ad essere utilizzato in ambito industriale, industriale a causa di collisioni tra pacchetti, ritrasmissioni, tempi di attesa casuali per poter accedere alla rete, che impedivano categoricamente un comportamento deterministico della comunicazione. É stato necessario attendere alcune innovazioni per rendere ethernet utilizzabile in ambito deterministico e Real Time.

Passiamo in rassegna questi problemi e le loro soluzioni:

- 100 Mb/s
  A causa dell'elevato numero di bit di overhead necessari al funzionamento del protocollo ethernet è stato necessario raggiungere la velocità di trasmissione di 100 Mb/s per ottenere un netto miglioramento rispetto ai bus di campo.

- Full-duplex
  É stato necessario sostituire lo standard half-duplex con il full-duplex per ridurre drasticamente la probabilità di collisione dei pacchetti ed eliminare il non determinismo introdotto dalle ritrasmissioni.

- Switch
  Un'altra innovazione volta a ridurre le collisioni è stata la sostituzione dei comuni HUB con i più evoluti switch, in grado di riconoscere il traffico e creare un canale comunicativo tra ogni dispositivo che lo richiede. Garantendo cosi la comunicazione immediata a ogni coppia di dispositivi.

Non appena questo livello tecnologico è stato raggiunto ogni azienda produttrice di bus di campo ha creato la propria versione in chiave ethernet del proprio protocollo di comunicazione.
In questo modo Siemens produttrice di Profibus ha creato ProfiNet, Allen Bradley da DeviceNet/-ControlNet ha introdotto EtherNet/IP, da Safety-BUS nasce SafetyNet etc.
Le differenze principali tra queste reti risiedono nell'architettura del sistema di comunicazione, nel modo in cui vengono evitate le collisioni tra pacchetti, nella costruzione del campo data del pacchetto ethernet e nella modifica o meno dello standard ethernet.

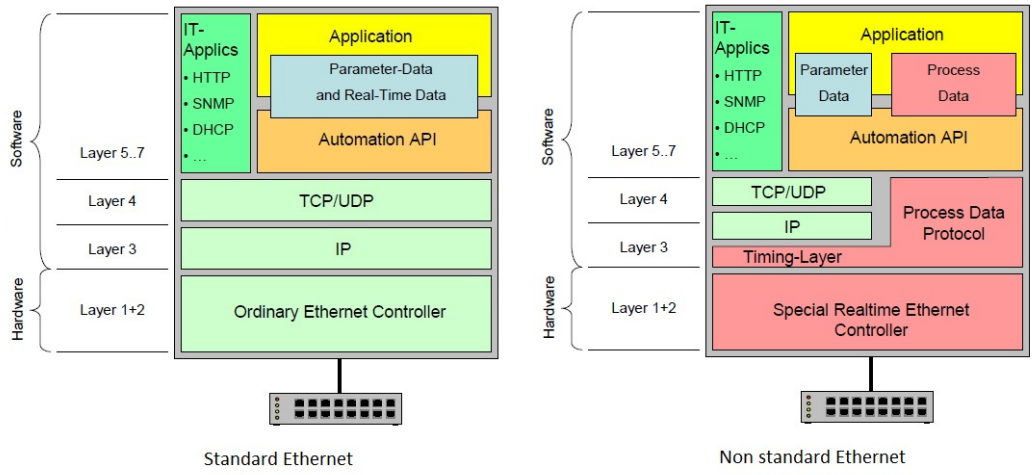


Fig. 2.3: Struttura standard ethernet e non standard ethernet

Come visibile in figura 2.3 con modifica allo standard ethernet si intende la sostituzione di uno o più livelli compresi tra 1 e 4 del modello ISO/OSI. In questo caso le modifiche non interessano i primi 2 livelli definiti dalla normativa IEEE 802.3 che determinano l'hardware su cui circolano i dati e come avviene l'accesso alla rete, ma si concentrano sui livelli 2 e 3, lo stack TCP/IP e UDP/IP cioè sul modo in cui i dispositivi scambiano i dati.
Essendo questi i protocolli maggiormente utilizzati della rete real-time ethernet, vengono brevemente descritti i tratti salienti:

- TCP (Trasmission Control Protocol)
  É un protocollo orientato alla connessione, utilizza dei pacchetti di servizio per stabilire

mantenere e chiudere la connessione tra mittente e destinatario; in questo modo garantisce l'arrivo a destinazione dei pacchetti, nell'ordine in cui sono stati spediti. Nella trasmissione viene richiesta una conferma da parte del ricevitore e in caso di errori sulla rete il pacchetto viene ritrasmesso.
A causa di questa struttura, il protocollo TCP non garantisce il determinismo nella comunicazione e quindi non viene utilizzato per comunicazione Real-Time ma solo per dati di parametrizzazione e diagnostica

- UDP (User Datagram Protocol) É un protocollo non connesso, i messaggi vengono spediti in rete senza garantire la consegna del pacchetto e l'ordine di arrivo, ma eliminando i messaggi di servizio si viene a creare una comunicazione più leggera e veloce rispetto a TCP. Per questo motivo viene preferito nelle comunicazioni Real-Time dove la ritrasmissione di un pacchetto perso non farebbe altro che peggiorare la performance della rete.

La modifica di questi due protocolli è pratica abbastanza comune dei produttori di reti industrial ethernet, che cosi facendo aumentano le performance della rete a scapito però della compatibilità diretta con le altre reti ethernet. Alcune esempi di queste reti sono: SERCOS III, EtherCAT, CC-Link IE, ProfiNet (IRT).
D'altra parte uno dei maggiori pregi delle reti industrial ethernet non modificate è la loro capacità di fondersi senza l'aggiunta di hardware specifico con le altre reti ethernet già presenti, come la gestione magazzino, risorse umane, acquisti-vendite etc. permettendo cosi il passaggio diretto di informazioni tra la rete di processo industriale a l'infrastruttura di ufficio sfruttando pienamente la semplicità di diagnostica e l'accesso rapido alle informazioni offerto dalla rete. In questo modo l'azienda riesce a migliorare i propri processi produttivi e risparmiare sull'infrastruttura di comunicazione.
Esempi pratici di questa tipologia di rete sono: EtherNet/IP, Modbus-IDA

Un'ultimo aspetto riguardate la connettività aumentata delle reti industrial ethernet è la gestione della sicurezza informatica, intesa come protezione dalle intrusioni non autorizzate sulla rete. Uno dei principali rischi di connettere la rete industriale alla rete di ufficio è infatti l'ingresso di persone estranee alla rete, che una volta ottenuto l'accesso possono danneggiare l'impianto o rubare informazioni riservate.
Diventa quindi importante predisporre strutture particolari come router e firewall per proteggere la propria infrastruttura.

Riassumendo i principali vantaggi di ethernet sui bus di campo sono:

- Maggior larghezza di banda
  É possibile trasmettere una quantità maggiore di dati, questo permette di utilizzare slave con intelligenza a bordo, decentrando parte del controllo e svolgendo attività di monitoraggio dei dispositivi stessi al fine di pianificare la manutenzione.
- É possibile migliorare le prestazioni temporali rispetto ai bus di campo.
- É possibile connettere un maggior numero di dispositivi e coprire un'area maggiore rispetto ai FieldBus.
- É possibile reperire materiali da vari produttori e a prezzi competitivi in quando ethernet è uno standard internazionale.
- Si semplifica il commissioning, della manutenzione, e dell'individuazione dei guasti.
- É possibile connettersi direttamente ad altre reti ethernet già presenti.

2.4 Wireless

Fig. 2.4: Impianto wireless

Un'altra tecnologia proveniente dal mondo commerciale che sta tentando di entrare nel modo industriale è la comunicazione senza fili wireless. In questo campo esistono molte tecnologie come il Wi-Fi con protocollo IEEE 802.11, Bluetooth, GPRS-EDGE, HSDPA, e non è ancora chiaro quale di queste sia la scelta migliore.
D'altra parte vi è una forte spinta a introdurre in ambito industriale questa nuova tecnologia, in quanto la possibilità di connettere i dispositivi senza fili, permette di abbattere i costi di messa in opera dell0impianto e di ampliare la possibilità di controllo a tutti quei dispositivi non raggiungibili fisicamente via cavo.
Esistono ancore numerosi problemi che devono essere risolti prima di introdurre massicciamente il wireless in ambito industriale:

- Costo
  Un sistema basato su wireless è più costoso perchè l'interfaccia wireless necessita di hardware particolare, e questo fa lievitare i costi di tutti i componenti dell'impianto.

- Sicurezza
  Le comunicazioni viaggiano in aria, un canale accessibile virtualmente a tutti. Quindi risulta necessario proteggere le comunicazioni con meccanismi di crittografia.

- Real-Time
  I tempi di accesso al mezzo trasmissivo sono molto più elevati rispetto al cavo, le operazioni di codifica e decodifica dei dati crittografati richiede hardware e tempo aggiuntivo per essere processati. Infine la probabilità di errore nella trasmissione sono molto più elevate.
  Tutto questo influisce negativamente sul determinismo della comunicazione e sulla velocità massima raggiungibile dalla rete.

Per questi motivi la reti wireless faticano a sostituire le reti cablate. Quindi sarà necessario attendere altre innovazioni tecnologiche perchè questo standard di comunicazione entri nel mondo industriale.

Capitolo 3

SERCOS

3.1 Nascita e sviluppo della rete SERCOS

La rete SERCOS III analizzata in questo elaborato è il frutto di circa 20 anni di ricerca e sviluppo da parte di varie associazioni del settore e finanziata dalle pi‘u importanti aziende tedesche e successivamente mondiali.
Nel 1987 la VDW, Associazione tedesca costruttori di macchine utensili forma con la ZVEI, Associazione tedesca industria Elettrica e elettronica un gruppo di lavoro per sviluppare una interfaccia aperta per sistemi di trasmissione digitali.
Nel 1989 alla fiera europea delle macchine utensili ad Hannover in germania viene presentata ufficialmente il frutto di questo lavoro cioè l'interfaccia di comunicazione SERCOS (SERial Time COmmunication System), un bus di campo orientato all'utilizzo negli azionamenti elettrici complessi quali macchine utensili multiasse, robot, e al dialogo con sensori e I/O.
Le peculiarità di questa interfaccia sono il mezzo fisico costituito da fibra ottica, la topologia di rete ad anello e la velocità di comunicazione fino a 4Mbit/sec.
Negli anni successivi viene fondato il gruppo FGS (utenti interfaccia SERCOS), diventano disponibili i primi ASIC per gestire la comunicazione con protocollo SERCOS e finalmente nel 1995 SERCOS diventa uno standard internazionale IEC 61491.
Successivamente alcune compagnie tra le quali spiccano alcuni nomi importanti:

ABB	Gildemeister	AEG
Index	AMK	Indramat
Baumueller	Siemens	Bosch

Formano il gruppo di interesse SERCOS per l'assistenza e la standardizzazione. Nel 1999 viene rilasciata la seconda versione di questa interfaccia denominata SERCOS II, le cui principali caratteristiche sono l'aumento della velocità di comunicazione a 16 Mbit/sec, la retrocompatibilità con SERCOS 1.
La terza generazione di SERCOS viene presentata nel 2005 e segna una svolta importante in questa interfaccia di comunicazione. SERCOS III è infatti una rete di comunicazione industriale che unisce i principi di funzionamento di SERCOS con il livello fisico e il protocollo di comunicazione Ethernet.
Le peculiarità di SERCOS 3 sono: velocità di comunicazione di 100 Mbit/sec, retrocompatibilità con le precedenti versioni per quanto riguarda i profili, la sincronizzazione, la struttura del messaggio e il set di parametri standard per descrivere gli aspetti del movimento in tempo reale.

3.2 Da SERCOS 1-2 a SERCOS III

SERCOS III è il classico esempio di come una rete possa cambiare il mezzo trasmissivo mantenendo inalterati, o quasi, tutti gli altri aspetti. Ma non solo, infatti alcune migliorie sono state apportate a vari livelli nella rete.
É stato introdotto il concetto di ridondanza fisica sulla rete, il quale permette di rilevare i guasti e ripristinare il collegamento tra il maggior numero possibile di slave e il master, a differenza di SERCOS I e II dove il danneggiamento di un cavo interrompeva la comunicazione su tutta la linea.
É stato implementato una struttura di comunicazione flessibile che permette a due slave di comunicare tra loro senza far intervenire il Master o cambiare la struttura dei messaggi. La medesima cosa è possibile fra due master collocati su reti diverse.
La comunicazione si basa ora sullo standard fisico di ethernet TCP/IP e non richiede l'utilizzo di hardware particolare, questo permette di:

- Risparmiare sui cavi, il cavo cat5e è più economico della fibra ottica
- Comunicare 6 volte più velocemente (100 Mb/s su sercos III contro i 16Mb/s su SERCOS II)
- Utilizzare programmi di monitoraggio reti (sniffer) come Wireshark.
- Commissioning dell'impianto via canale IP, senza obbligo di sottostare al protocollo SERCOS III

Grazie a queste caratteristiche offerte dal livello fisico di ethernet full duplex, SERCOS III ha potuto diminuire il proprio tempo minimo di ciclo arrivando a chiudere l'anello logico in 31.25 μs contro i 62.5 μs di SERCOS I e II.
Un'ultima innovazione è la fusione in un'unico bus delle comunicazioni safety e standard, infatti acquistando controllore e azionamenti certificati safety (Safety on Board) è possibile, senza modificare la topologia di rete scelta, trasmettere messaggi legati alla sicurezza dell'impianto sulla stessa rete dove è presente il normale traffico.
Tutti questi cambiamenti hanno però mantenuto un cero grado di omogeneità tra le varie versioni:

- I profili Servo e Motion sono rimasti compatibili
- Il software a livello applicazione è rimasto compatibile
- L'utilizzo del canale di servizio non è stato modificato
- le prestazioni real time e di sincronizzazione sono rimaste inalterate e in alcuni casi migliorate

Nella tabella 3.1 vengono riassunti i principali cambiamenti introdotti nelle varie versioni di SERCOS.

	SERCOS 1	SERCOS 2	SERCOS 3
Data	1987	1999	2005
Mezzo fisico	fibra ottica	fibra ottica	Ethernet
Topologia rete	Anello	Anello	Lineare o Anello
Velocità di trasmissione	2/4 Mb/s	2/4/8/16 Mb/s	100Mb/s
tempo di ciclo	Configurabile, minimo 62.5μs	Configurabile, minimo 62.5μs	Configurabile, minimo 31.25μs
Jitter	< 1 μs	< 1 μs	< 1 μs
Sincronizzazione	Sincronizzazione hardware		
Protocollo base	HDLC		Ethernet
Protocollo Real Time	SERCOS		
Ridondanza Hardware	No	No	Si con topologia ad anello
Comunicazione incrociata diretta	No	No	Si
Comunicazione e sincroniz. tra sistemi di controllo	No	No	Si
Canali di servizio	No	No	Si
Canale opzionale NRT	No	No	Si
Hot-plugging	No	No	Si
Numero di Master	1 per anello	1 per anello	1 per anello/linea
Numero di nodi	254 per anello, più anelli possibili	254 per anello, più anelli possibili	254 per anello/linea, più anelli/linee possibili

Tab. 3.1: Tabella riassuntiva

3.3 Funzionamento fisico di SERCOS III

A livello fisico SERCOS III utilizza la tecnologia fast ethernet per la comunicazione a 100 MB/s tra i dispositivi. Gli standard supportati sono IEEE 802.3 & ISO/IEC 8802-3, 100Base-TX (fast ethernet su coppia di cavi cat5) oppure 100Base-FX (Fast ethernet su fibra ottica) tutti operanti in full duplex.
La comunicazione a livello hardware è gestita tramite un FPGA o soluzioni del tipo Gate-Array, in commercio il chip Xlinx Spartan-3 e Altera Cyclone II sono in grado di gestire il protocollo SERCOS III, mentre il chip Hilscher netX è un controller multiprotocollo per ethernet che fra i vari protocolli supportati annovera anche SERCOS III.
Circa 60 produttori distribuiti in tutto il mondo supportano nei loro prodotti lo standard SERCOS 3; questo rende la rete aperta e permette di avere ampia scelta e concorrenza sul materiale da utilizzare per costruire l'impianto desiderato. Sercos ha due gruppi di lavoro, uno in Europa e uno in Nord America, questo permette di adattare i propri standard di sviluppo in base alle esigenze locali. Ogni dispositivo presente su una rete SERCOS III è caratterizzato dalla presenza di due interfacce ethernet full duplex chiamate porta 1 (P1) e porta 2 (P2). Le due porte non presentano differenze tra loro, e possono essere utilizzate in maniera intercambiabile, semplificando il cablaggio e riducendo il rischio di errori.
Su ciascuna porta ethernet full duplex è inoltre possibile individuare due Physical Medium Attachment (PMA), attraverso i quali si possono instaurare due canali di comunicazione uno chiamato primario e l'altro secondario.

La struttura topologica della connessione tra i dispositivi in SERCOS III è limitata a solo due forme schematizzate in figura 3.1:

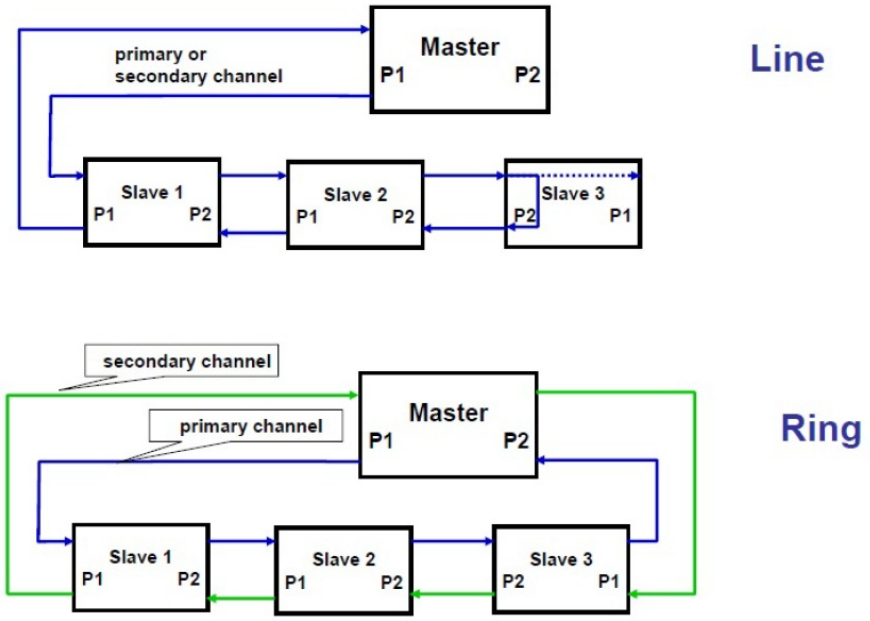


Fig. 3.1: Topologia lineare e ad anello

- Topologia lineare :
  La più semplice da implementare, si risparmia un cavo, ma si perde la ridondanza nelle comunicazioni.
  In questa configurazione il master utilizza una sola porta per la connessione verso gli slave, i quali a loro volta sono connessi tra loro uno di seguito all'altro a formare una linea. L'ultimo dispositivo è caratterizzato dall'avere una sola porta impiegata per la connessione verso la rete, la porta inutilizzata rimane attiva e viene continuamente monitorata dallo slave per fornire il servizio di hot plug-in che verrà analizzato successivamente.
  In questa topologia si crea un anello virtuale sui dispositivi nel seguente modo: il master inizia la comunicazione spedendo un telegramma attraverso la propria porta attiva (ognuna delle due può esserlo) al primo slave, lo slave riceve il pacchetto attraverso una delle sue porte, lo legge, eventualmente apporta delle modifiche e lo inoltra allo slave successivo l'altra porta.
  Questa procedura continua di slave in slave fino a che il pacchetto non raggiunge l'ultimo dispositivo della linea; l'ultimo slave non avendo altri dispositivi attaccati alla sua seconda porta, processa il pacchetto e lo spedisce indietro allo slave precedente. Il pacchetto ripercorre nuovamente la rete nel senso inverso e raggiunge il master completando cosi il suo tragitto.

- Topologia ad anello:
  La topologia ad anello si ottiene a partire dalla topologia lineare semplicemente aggiungendo una connessione tra l'ultimo dispositivo presente sulla rete e la seconda porta del master. La comunicazione avviene qui in maniera leggermente diversa rispetto alla configurazione lineare, non appena il master comprende di trovarsi in un anello crea due telegrammi identici e li spedisce ai dispositivi connessi alle sue due porte. Ognuno di questi due pacchetti controrotanti attraversano la rete nello stesso modo della topologia lineare e terminano alla

porta opposta del master.
Questa topologia è intrinsecamente ridondante, come è possibile vedere in figura 3.2 la rottura di un cavo o un dispositivo non isola una parte della rete, che rimane sempre raggiungibile dall'altra porta del master. Inoltre appena il master, non ricevendo i pacchetti attesi identifica un malfunzionamento nella rete, attiva una procedura di recovery. Attraverso di essa e senza creare collisioni apporta modifiche alla comunicazione in modo da raggiungere quanti più dispositivi possibile creando di fatto due nuova reti lineari, una per ciascuna delle due porte. Tutto questo in un tempo inferiore a 25 μs, inferiore quindi anche al tempo di ciclo minimo, questo comporta un ripristino delle funzionalità della rete senza perdita di dati o l'interruzione delle comunicazioni.

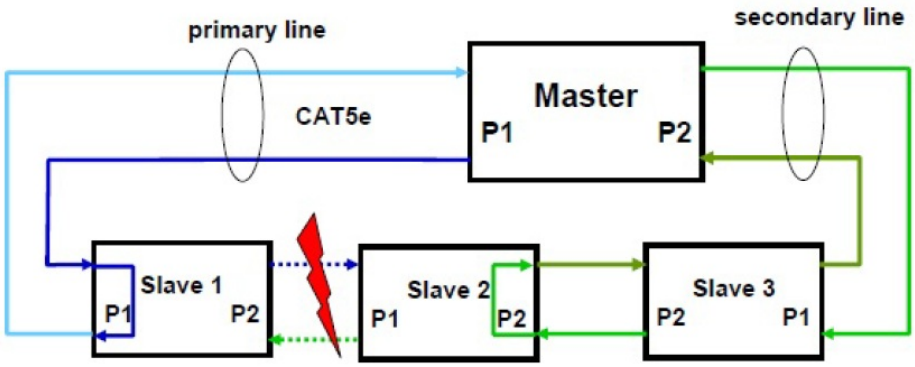


Fig. 3.2: Ridondanza in rete ad anello

Ogni altra topologia è espressamente non supportata, come non sono ammessi nella rete logica dispositivi come switch o HUB.
Sercos III è stato progettato specificatamente per supportare un controllo distribuito dei dispositivi, utilizzando dei driver e sensori con intelligenza a bordo per sgravare il controllore centrale di parte del carico di lavoro necessario all'elaborazione dei dati provenienti dal campo. E ad oggi è l'unica rete che supporta il controllo concentrato e distribuito, realizzando una maggior efficienza e flessibilità della rete.
Il numero massimo di dispositivi che un master può sopportare nella stessa rete è 511, ma questo limite è facilmente superabile utilizzando più di un master e sfruttando la peculiarità della comunicazione controller-to-controller SERCOS III.
La connessione tra i dispositivi sulla rete può avvenire con cavi schermati CAT5e S/UTP in cui viene connesso a massa lo schermo e tutti i fili non utilizzate come consigliato da Bosch-Rexroth, oppure con cavi più economici come i classici cat5e crossover o patch, a scapito ovviamente della schermatura contro i disturbi. In ogni caso la lunghezza massima dei cavi utilizzati deve rimanere inferiore ai 100 metri. I connettori possono essere del tipo RJ45 o M12 con un grado di protezione IP adeguato all'ambiente di lavoro (IP20 o IP 67).

-Meccanismo di hot-plugin
L'ultimo dispositivo presente nella rete a topologia lineare oltre a rispedire i pacchetti verso il master, li invia alla sua seconda porta anche se ad essa non è connesso nessun dispositivo. Questo per permettere ad un eventuale dispositivo appena connesso alla rete e non configurato di iniziare a comunicare con il master. Questa procedura è sempre attiva e non richiede il fermo o il reset della rete stessa.

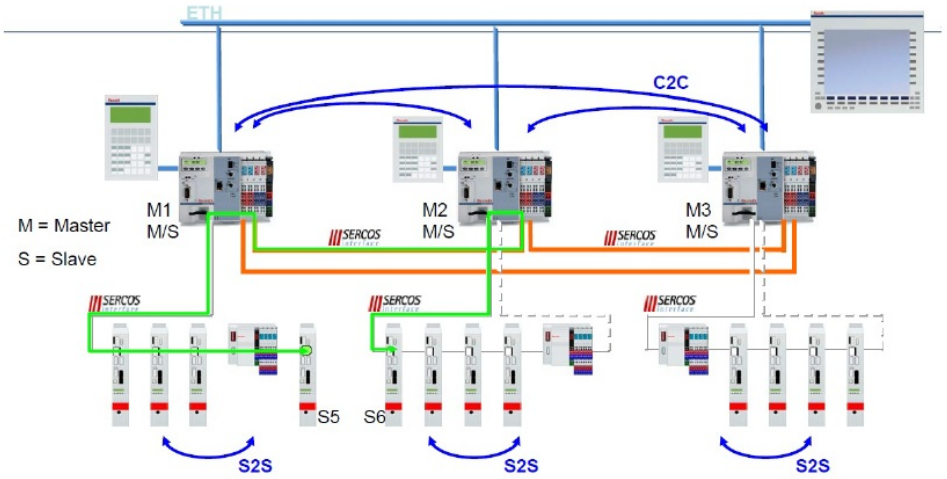


Fig. 3.3: Comunicazione controller-to-controller e slave-to-slave

Nel momento in cui un nuovo dispositivo richiede l'accesso alla rete si susseguono 3 fasi che portano alla modifica dell'anello logico per permettere l'ingresso di questo nuovo dispositivo. Una disegno schematico è visibile in figura 3.11

- Fase 0 HP (viene utilizzato il campo hot-plugin)
  Il master spedisce gli stessi parametri a tutti gli slave che cercano di entrare nella rete, successivamente il master controlla l'indirizzo del primo slave che desidera entrate e lo porta in modalità parametrizzazione slave.
- Fase 1 (viene utilizzato il campo hot-plugin)
  Il master trasmette i diversi parametri ai vari dispositivi che cercano di entrare in rete, successivamente commuta la comunicazione sul canale di servizio.
- Fase 2 (viene utilizzato il canale di servizio)
  Il master trasmette i parametri necessari per configurare completamente lo slave e non appena questa fase è conclusa inizia la comunicazione real-time dei dati.

3.4 Funzionamento logico di SERCOS III

Si analizza ora nello specifico come i pacchetti circolanti in rete possono trasferire informazioni, e nello specifico come SERCOS III risolve il problema del determinismo in ethernet.
Per prima cosa tutti i messaggi circolanti in rete sono indirizzati a tutti i dispositivi presenti in rete (indirizzo di Broadcasting: ff ff ff ff ff ff), e non facendo uso dell'indirizzo ip nella comunicazione l'indirizzo del mittente è vuoto (00 00 00 00 00 00).
Questo non deve far pensare che dispositivi presenti in rete non siano identificabili singolarmente, perchè in SERCOS III sono presenti ben due tipi di indirizzamento:

- SERCOS III address: Un numero intero compreso tra 1 e 511 identifica in maniera univoca ciascun dispositivo presente nella rete e viene utilizzato come un indice dagli altri dispositivi per comunicare specificatamente con esso.
- IP address: Non utilizzato da SERCOS III per le proprie operazioni, la sua presenza può essere necessaria per supportare altre specifiche o altre comunicazioni parallele a SERCOS III

L'unica precauzione da adottare è l'utilizzo di particolari dispositivi come i router per limitare la diffusione dei pacchetti SERCOS III oltre i confini desiderati.
Il modello gerarchico che sta alla base della comunicazione in SERCOS III è il classico schema master-slave, dove il master inizia e termina ogni comunicazione. In questo modo viene eliminata di fatto la possibilità che due messaggi vengano messi in rete contemporaneamente andando a scontrarsi.
Osservando poi la figura 3.4, che mostra una schematizzazione del traffico SERCOS III, è facile notare la struttura ciclica della comunicazione dove si alternano in successione il MDT (Master Data Telegram), l'AT (Acknowledge Telegrams) e l' IP Channel detto anche NRT (Non Real Time Channel). Uno o più telegrammi MTD seguito da uno o più AT formano un ciclo che è il mattone fondamentale su cui si basa lo scambio di informazioni in SERCOS.

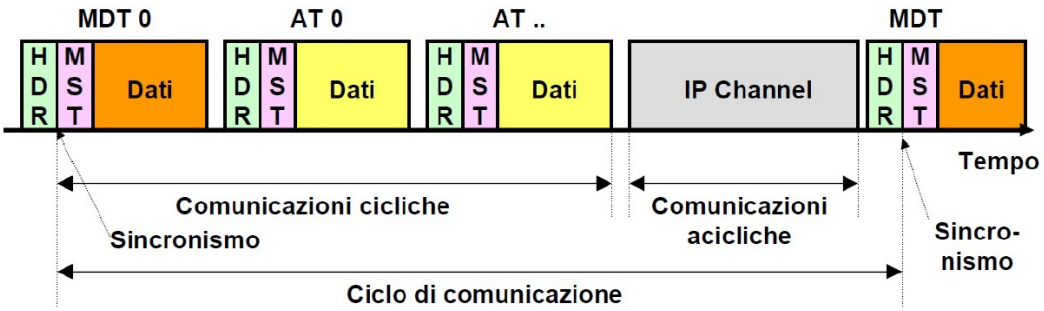


Fig. 3.4: Struttura comunicazione SERCOS III

Per rendere più chiaro il meccanismo di scambio dati, viene di seguito descritto un esempio semplice di comunicazione:
Per prima cosa il master spedisce in rete il telegramma MDT, che a seconda delle dimensioni può essere suddiviso fino a 4 frame ethernet consecutivi. Questo messaggio contiene principalmente le informazioni indirizzate agli slave per il corretto funzionamento del processo.
Essendo la comunicazione di tipo broadcasting, il frame raggiunge tutti i dispositivi presenti sulla rete, i quali estraggono le informazioni che li riguardano e ignorano quelle destinate agli altri dispositivi. Successivamente il master invia un nuovo messaggio, il telegramma AT che come il precedente può essere suddiviso fino a 4 pacchetti ethernet consecutivi, ma a differenza del MDT non contiene dati, cioè si presenta vuoto nel momento della creazione. Nel suo viaggio lungo la rete, ogni dispositivo che attraversa aggiunge al pacchetto i propri dati, in modo che una volta raggiunta la fine della rete il pacchetto AT sia completo e possa portare al master le informazioni di tutti gli slave in un unico telegramma.
Nell'analisi appena fatta sorge il problema di selezionare nel messaggio condiviso l'intervallo di byte destinati a ogni singolo slave. SERCOS III risolve questo problema con l'utilizzo di indici (SERCOS III addressing). Questi indici vengono assegnati dal master a ogni singolo slave nel momento in cui la rete viene inizializzata, cosi tutti i messaggi che circolano in rete vengono interpretati come una serie di dati, e ogni dispositivo può accedere al campo che lo interessa attraverso questo indice.
Il controllo dell'integrita dei dati comunicati avviene tramite il Frame Check Sequence (FCS) da parte di tutti i dispositivi che ricevono un pacchetto.
Nel caso del MDT il master crea il pacchetto con i dati, e genera il FCS, ogni slave che lo riceve per prima cosa controlla il FCS e in questo modo conferma o meno l'integrità dei dati. Nel caso del AT, ogni slave che riceve il pacchetto controlla il FCS come nel MDT, e una volta aggiunti i propri dati al pacchetto calcola il nuovo FCS e lo sostituisce al precedente e spedisce il nuovo pacchetto. In questo modo ogni comunicazione tra due dispositivi qualsiasi viene controllata. Oltre

alla struttura di comunicazione classica Master-Slave, per ottenere migliori risultati nel controllo, SERCOS III ha sviluppato delle comunicazione drive-to-drive (D2D), in modo che i dispositivi slave possono comunicare tra loro senza passare per il master che aggiunge ritardi di comunicazione e overhead. Questo principio è stato poi esteso al caso Controller-to-Controller (C2C) dove i controllori possono comunicare tra loro su diverse reti nello stesso sistema. Si è giunti infine a un metodo standard per lo scambio di informazioni a livello di controllo e sincronizzazione nei tre casi.
In SERCOS III è quindi supporta sia una strategia di controllo accentrata dove si chiudendo sul master tutti gli anelli di controllo, sia strutture ibride dove l'anello di corrente è chiuso sul driver e gli anelli di velocità e posizione chiusi sul controllore centrale.
Nel tempo rimanente tra la ricezione da parte del master del AT completo e l'inizio del nuovo ciclo, la rete può essere utilizzata per la trasmissione di dati non ciclici da e verso i dispositivi (IP channel). Questa funzione è prevalentemente utilizzata per scopi diagnostici, ma nulla vieta che altri protocolli possono accedere alla rete per comunicazioni non legate al processo da controllare. Questo fa di SERCOS III una rete aperta, dove più protocolli possono coesistere senza hardware particolare o incapsulamento dei dati (tunneling) e svolgere le loro comunicazioni.

3.5 Frame SERCOS III

Si procede analizzando i campi che compongono un frame ethernet di tipo SERCOS III.

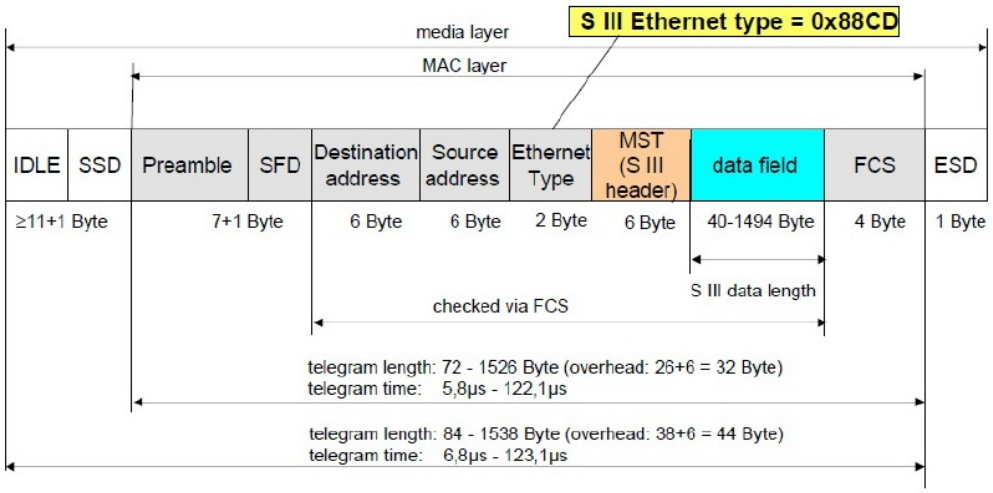


Fig. 3.5: SERCOS III Telegram

- Preamble:
  É una sequenza di 7 byte e servono a svegliare gli adattatori del ricevente e a sincronizzare gli oscillatori con quelli del mittente.

- SFD:
  Start Frame Delimiter formato da un byte, è utilizzato per informare il destinatario che sta arrivando del contenuto importante nel messaggio.

- Destination Address:
  In SERCOS III tutti i pacchetti che trasportano dati ciclici sono indirizzati a tutti i dispositivi presenti sulla rete (Broadcast), quindi in questo campo è presente l'indirizzo di broadcasting ffff.ffff.ffff (HEX).

- Source Address:
  La rete SERCOS III non utilizza l'indirizzo IP per svolgere le proprie operazioni. Quindi in tutti i pacchetti che trasportano dati ciclici questo campo è sempre 0000.0000.0000 (HEX).Questo non esclude che i dispositivi abbiano lo stesso un proprio indirizzo IP utilizzato nella comunicazione con altri protocolli.

- Ethernet Type:
  Questo campo formato da due byte contiene un codice che indica il tipo di protocollo in cui è incapsulato il campo dati (data field). Nel caso in esame l'ethertype SERCOS III è 88cd (hex).

I campi analizzati fino ad ora fanno parte dello standard ethernet. Procedendo nel pacchetto si incontra il campo dati, dove si analizza a basso livello il funzionamento di SERCOS III. L'inizio del campo dati del frame ethernet SERCOS III è chiamato header (o Master Set Telegram MST) ed è formato da 2 byte che contengono le informazioni necessarie alla sincronizzazione dei dispositivi con il master. Questo campo si interpreta in binario invertendo l'ordine dei due byte che compongono la word:

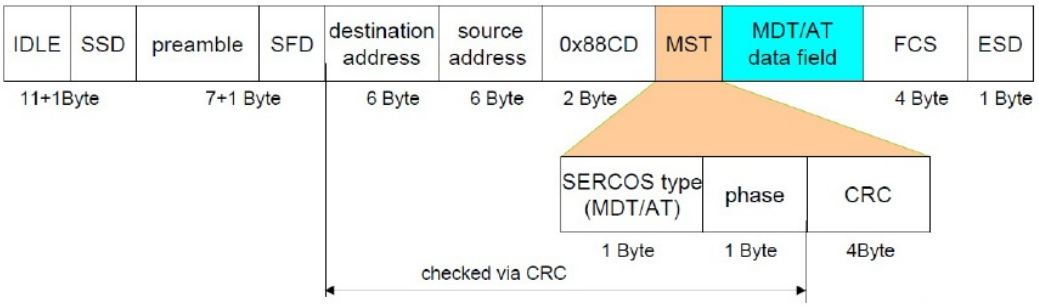


Fig. 3.6: Master Set Telegram

- 1° byte: Telegram Type:
  bit 15: indica il canale di comunicazione: 0 = P-telegram, telegramma sul canale primario, 1 = S-telegram, telegramma sul canale secondario
  bit 14: indica il tipo di telegramma: 1 = AT (acknowledgement telegram), 0 = MDT (Master Data Telegram).
  bit 13: indica la validità del ciclo: 1= Valid, 0 = invalid
  bit 12: Riservato.
  bit 11,10,9,8: indica la numerazione dei frame riguardanti lo stesso telegramma. Se il MDT o AT vengono suddivisi in pi'u frame, questo campo riporta il numero dell'attuale frame.

- 2° byte: Phase
  bit 7,3,2,1,0: indica numericamente in che fase si trova la rete di comunicazione (si veda il paragrafo inizializzazione della rete)
  bit 6,5,4: indica numericamente a che ciclo di comunicazione appartiene il pacchetto. I cicli sono numerati in ordine crescente con un meccanismo di overflow una volta raggiunto il numero massimo di 7.

- Codice ridondanza ciclica (CrC32):
  Formato da 4 byte viene utilizzato per garantire la correttezza della trasmissione dei dati

contenuti nei campi:indirizzo destinazione, indirizzo sorgente, EtherType, SERCOS type, Phase.

Il successivo elemento del pacchetto è il campo dati SERCOS III, formato al massimo da 1500 byte contiene un set di variabili configurabili per ogni dispositivo presente nella rete. Questo campo può essere ulteriormente suddiviso in diverse parti:

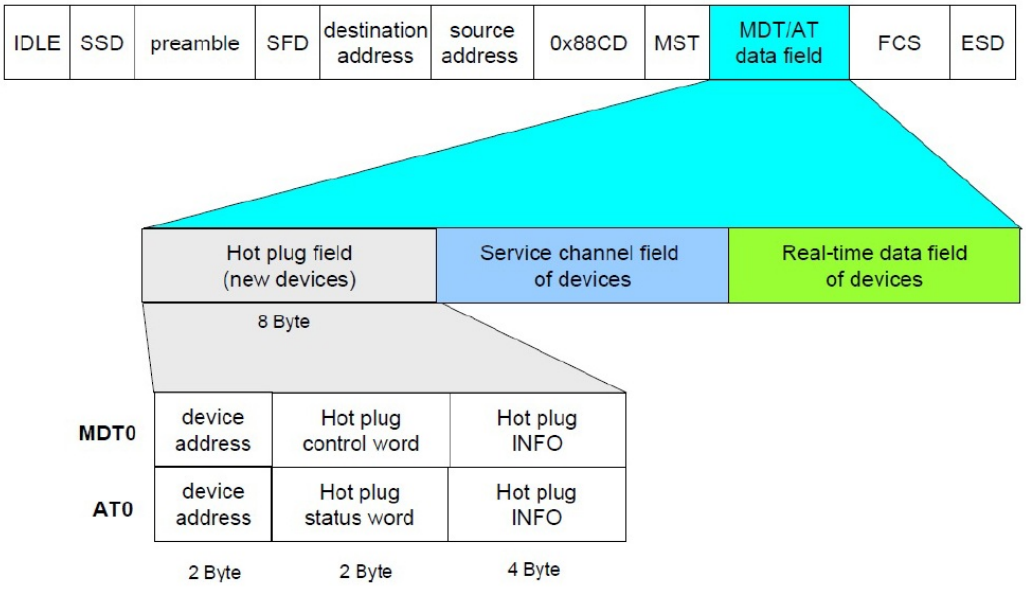


Fig. 3.7: Hot Plug

- Hot Plug:
  Questo campo formato da 8 byte è presente all'inizio del campo dati solo nel primo frame del MDT e AT e solo nelle fasi di comunicazione 3 e 4. Può essere a sua volta suddiviso in tre sotto-campi:
  2 byte: indirizzo del dispositivo
  2 byte: hot-plug control/status word 4 byte: hot-plug info

- SVC 1...n
  Successivamente nel campo dati si trova il campo dedicato al canale di servizio (SerVice Channel), dove ad ogni dispositivo sono riservati 6 byte consecutivi per le comunicazioni extra a quanto concordato con il master in fase di inizializzazione della rete. Un trasferimento su canale di servizio può durare più di un ciclo di comunicazione.

- Control device/ Status device
  Questo campo formato da 2 word (di cui una sola utilizzata) contiene dei comandi particolari come l'accensione-spegnimento dei dispositivi piùttosto che la segnalazione di un malfunzionamento nel dispositivo.
  Si differenziano a seconda che il messaggio a cui appartiene sia il MDT o il AT, prendendo il nome rispettivamente di control word e status Word. Vanno interpretate in binario e lette da sinistra verso destra con il seguente significato:

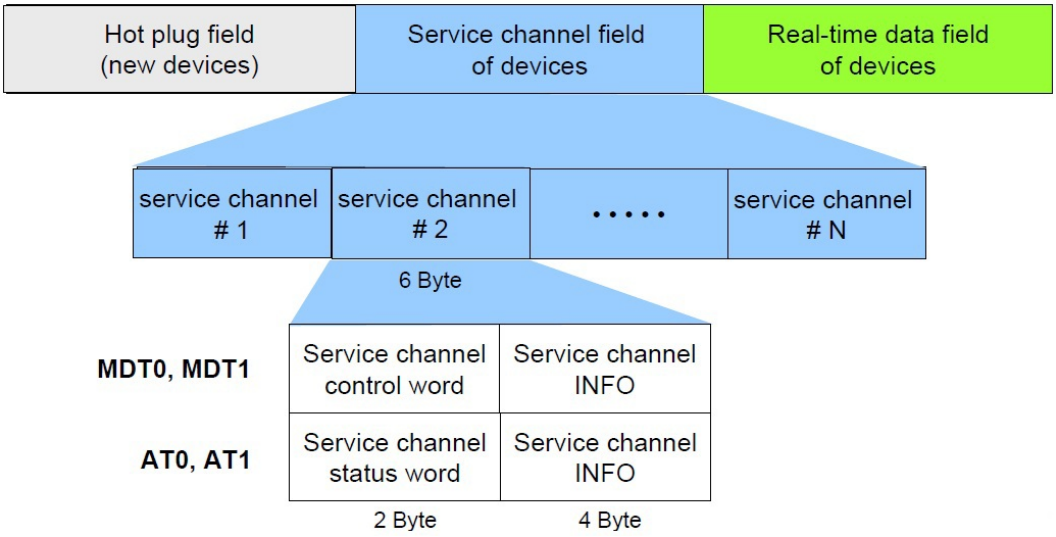


Fig. 3.8: Campo service channel

CONTROL WORD
Bit 5 - 0:Controllo informazioni del canale di servizio

Bit 7 - 6: Bits controllo Real-Time 1 e 2. Sono configurabili dall'utente della rete secondo le sua necessità.

Bit 9 - 8: Modalita comando
-00: modalità primaria
-01: modalità ausiliaria

Bit 10: IPOSYNC: impulso di interpolazione, commuta se ci sono nuovi valori di comando da trasmettere

Bit 12: Riservato

Bit 13: Drive HALT, la transizione 1 $\Rightarrow$ 0, mette l'unità in stato di STOP, pur restando sotto controllo. (possibile solo se i bit 14 e 15 sono a 1)

Bit 14: Drive ENABLE, la transizione 1 $\Rightarrow$ 0, toglie coppia immediatamente all'azionamento (indipendenti dal bit 15 o 13)

Bit 15: Drive ON, la transazione 1 $\Rightarrow$ 0, spegne l'azionamento nel miglior modo possibile (a seconda della configurazione dell'azionamento), l'azionamento può restare alimentato.(possibile solo se il bit 14 'e a 1)

STATUS WORD

Bit 2 - 0: Riservati

Bit 3: Stato di trasferimento del riferimento.
0 = Il driver ignora il riferimento
1 = Il driver segue il riferimento

Bit 4: Drive halt
0 = drive halt non attivo
1 = drive halt attivo

Bit 5: Stato del valore posizione attuale (bit 0 del IDN S-0-0403.0.0)

Bit 7 - 6: Bit di stato Real-Time 1 e 2. Configurabili dall'utente della rete secondo le sue necessità.

Bit 9 - 8: Tipo di operazione attuale
-00: Operazione in modalità primaria
-01: Operazione in modalità secondaria

Bit 11: Cambio bit diagnostica classe 3. La commutazione segnale che 'e disponibile un nuovo dato per il master.

Bit 12: Cambio bit diagnostica classe 2
0 = non abilitato
1 = cambio

Bit 13: Driver bloccato, errore in diagnostica classe 1
0 = nessun errore
1 = Driver bloccato da un errore

Bit 15 - 14: Pronto ad operare:
-00: Driver non pronto per l'accensione a causa di connessioni interne non positivamente collegate.
-10: Controllo di alimentazione pronto per il funzionamento, l'azionamento non è in coppia.
-01: Pronto per l'accensione.
-11: In funzionamento, l'azionamento è in coppia.

- campo dati:

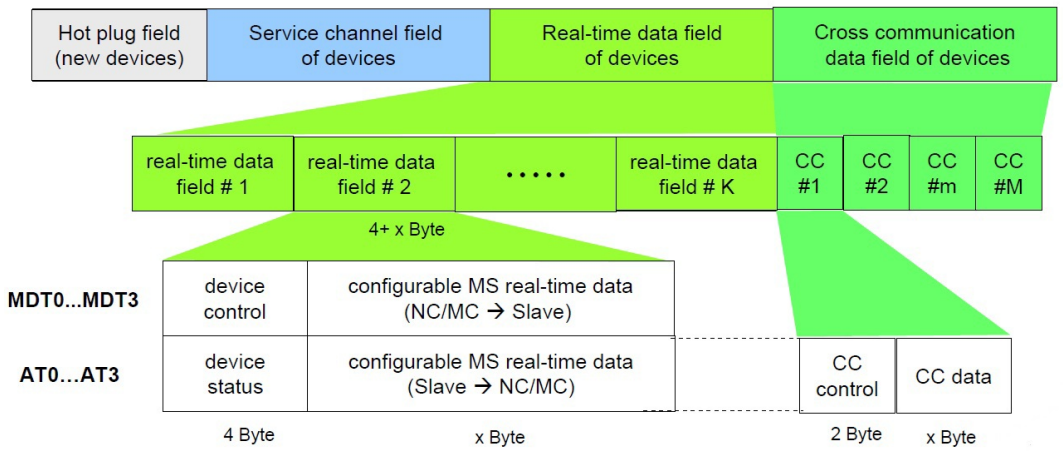


Fig. 3.9: Campo dati

Appena dopo lo status word/control word troviamo i dati destinati a ogni singolo dispositivo, la struttura e l'ordine di questi dati viene decisa in fase di inizializzazione della rete e dipende dal tipo di azionamento e di controlli che si intende utilizzare.

- FCS:
  Frame Check Sequence, si tratta di un altro controllo sulla integrità dei dati trasmessi; viene calcolato dal master ed eventualmente modificato dagli slave che rielaborano il messaggio.

- IFG:
  Interframe Gap si tratta di bit aggiuntivi, che non trasportano informazioni, ma servono per rendere il frame sufficientemente lungo per essere trasmesso secondo lo standard ethernet.

3.6 Sincronizzazione della rete

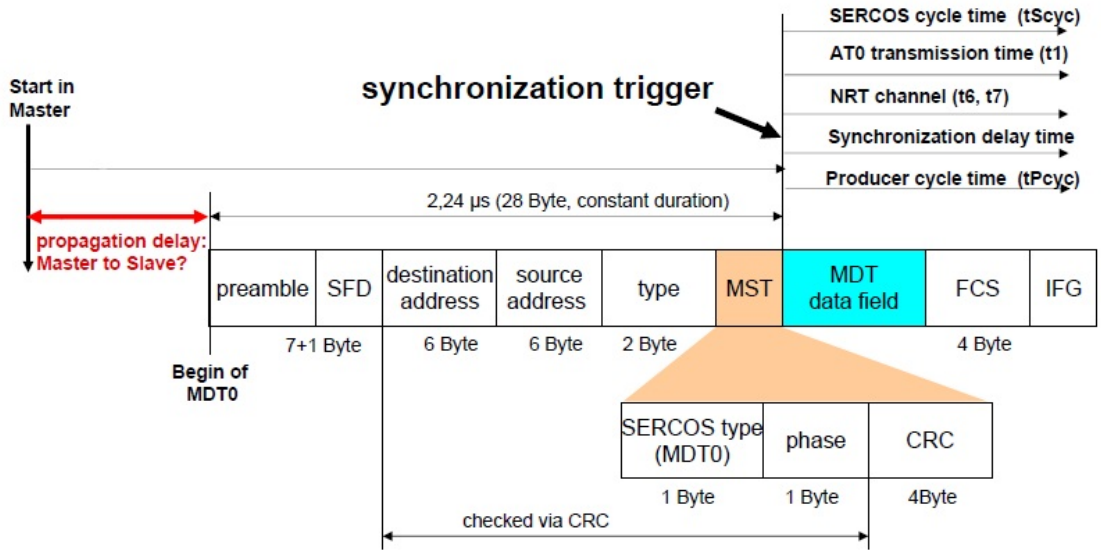


Fig. 3.10: Sincronizzazione SERCOS III

A differenza di altre reti di comunicazione, come EtherNet/IP che si affidano al protocollo IEEE 1588 per mantenere tutti i dispositivi presenti in rete sincronizzati tra loro, SERCOS III affida questo compito alla struttura particolare delle comunicazione SERCOS, sfruttando il fatto di utilizzare un proprio protocollo modificato di ethernet.
Questa sincronizzazione è affidata principalmente al master, che attraverso il primo MDT (MTD0) di ogni ciclo di comunicazione crea una riallineamento dei tempi di tutti i dispositivi presenti in rete. Quindi la precisione della rete dipende principalmente dal clock del master e dalle misure del ritardo di comunicazione della rete.
Affrontiamo ora il problema di calcolare i ritardi di propagazione della rete e dei dispositivi.
Il ritardo imputabile al cavo in rame CAT5e è al massimo 5.56 *etas*/m, mentre il ritardo introdotto dalla fibra ottica è 5 *etas*/m che comportano un ritardo di rispettivamente 556 *etas* e 500 *etas* su 100 metri di cavo. Il ritardo imputabile ai dispositivi, dipende invece dal produttore e dal tipo di FPGA utilizzato per gestire il protocollo di comunicazione, vari esempi vengono analizzati nella seguente tabella:

PHYRx	PHYTx	dispositivo	marca
220 ηs	90 ηs	KS8721BL	Micrel
215 ηs	60 ηs	DP838481	NCS
170 ηs	50 ηs	LXT973	Cortina
600 ηs	600 ηs	netX	Hilsher

Il ritardo totale di trasmissione viene quindi calcolato dal master sommando i vari ritardi come visibile in figura 3.11

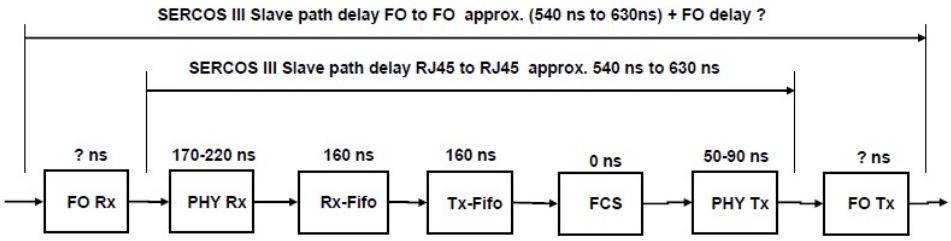


Fig. 3.11: Ritardo totale introdotto in una comunicazione SERCOS III

E trasmesso a ogni slave in fase di inizializzazione della rete, in modo che nella successiva fase 4 di comunicazione le informazioni vengano spedite e ricevute in maniera più precisa possibile.
Nel caso in cui la topologia utilizzata sia invece ad anello, vengono spediti contemporaneamente dal master sulla rete due telegrammi controrotanti, che raggiungono gli slave in tempi diversi, in questo modo ogni slave può scegliere quale il telegramma con minor ritardo e sincronizzarsi con esso.

3.7 Inizializzazione della rete SERCOS III

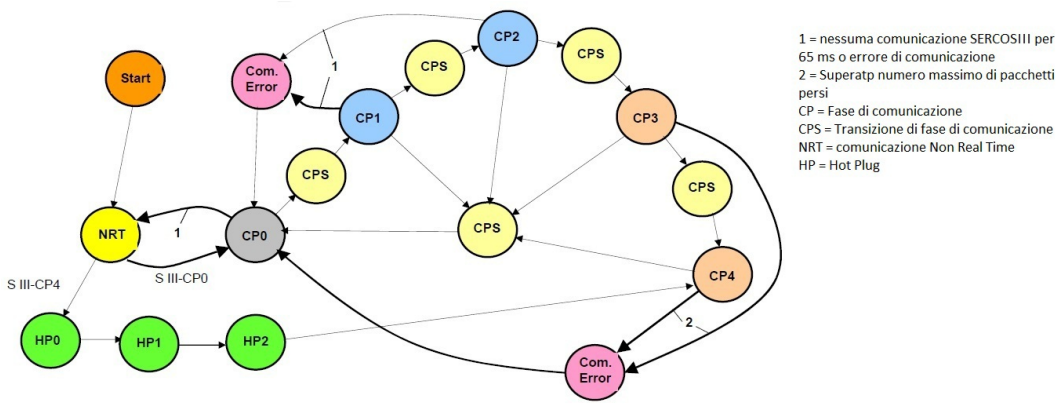


Fig. 3.12: Schema delle fasi di accensione della rete SERCOS III

Nel paragrafo precedente si è analizzato un generico pacchetto SERCOS III, ma alcuni campi come i dati degli slave restano incomprensibili, perchè la loro struttura viene decisa nelle fasi immediatamente successive all'accensione dei dispositivi presenti in rete.
Per meglio comprendere il contenuto di questi campi e la strategia di comunicazione SERCOS III si analizza ora lo start-up della rete.
Il primo avvio dell'impianto detto commissioning, inizia dalla parametrizzazione off-line dei singoli dispositivi. Per fare questo Bosch-Rexroth mette a disposizione alcuni programmi, in questa sede si è utilizzato il software Indraworks mlc04 v22. Una volta costruito virtualmente l'impianto

inserendo tutti i codici identificativi dei dispositivi è possibile connettersi fisicamente all'impianto. Il passaggio da uno stato al successivo è sempre comandato dal master e preceduto da una serie di controlli su alcuni dati sensibili. Si analizza ora ogni singola fase:

- FASE 0: Verifica della topologia della rete, scambio dati aciclico
  In questa fase il Master spedisce sulla rete solo dei pacchetti MDT, questi pacchetti non contengono dati, servono solo a verificare l'integrità delle connessioni e la capacità dei dispositivi di inoltrare un pacchetto. Gli slave in questa fase si comportano come dei repeater, si limitano a ricevere ogni MDT e spedirlo allo slave successivo.
  Dopo che il master ha ricevuto un adeguato numero di pacchetti che hanno compiùto un intero giro della rete e quindi verificato la topologia e l'integrità dell'anello logico comanda il passaggio a fase 1.

- FASE 1: Verifica dispositivi sulla rete
  In questa fase il master spedisce in rete sia dei MDT che degli AT. I MDT non contengono dati per gli slave, ma solo una richiesta di hanshake nel canale di servizio. Gli slave interrogati rispondono a loro volta confermando che il canale di servizio è valido e rispondendo all'handshke. Dopo un certo numero di risposte da parte degli slave il master comanda il passaggio a fase 2.

- FASE 2: Parametrizzazione della rete
  In questa fase il master è a conoscenza degli slave presenti sulla rete e comincia a istruire ogni singolo dispositivo attraverso gli Identification Numbers (IDNs), dei particolari codici standardizzati da SERCOS che permettono di definire ogni aspetto della comunicazione e del tipo di dati che gli slave possono utilizzare. Una esaustiva spiegazione degli IDNs verrà fornita nel prossimo paragrafo. In questa fase viene definito in particolare il tempo di ciclo, la composizione della lista di dati nel MDT e AT, viene controllata che la parametrizzazione off-line sia corretta etc.

- FASE 3: Inizio comunicazione dati ciclica
  A partire dalla fase 3 la rete comincia a comunicare in Real-Time secondo i tempi e i modi definiti in fase 2. In questa fase vengono ultimati i controlli sulla rete. Inoltre, in questa fase viene calcolato il ritardo di comunicazione nodo-nodo misurato dall'accensione della rete fino ad adesso, in modo che questo venga compensato nel funzionamento della rete in fase 4.

- Fase 4 La rete è completamente parametrizzata e testata. É possibile mandare in esecuzione i programmi e scambiare messaggi in real-time secondo quanto parametrizzato in fase 2 e 3. La rete rimane in fase 4 fino a che si desidera utilizzare la parametrizzazione e il programma scelto, per ogni cambiamento hardware o software alla rete è necessario ritornare in fase 2.

3.8 IDNs comunicare con gli identificatori

In SERCOS tutti i dati di parametrizzazione dell'impianto, come i fattori di scala, i guadagni di anello e i tempi di ciclo, e tutti i dati riguardanti il funzionamento del processo come posizione attuale motori, velocità attuale motori, sensori di I/O, sono stati standardizzati in un formato particolare chiamato IDN (IDentification Number). La struttura utilizzata in IEC per la descrizione degli IDN è riportato nell'immagine: 3.12
Nello specifico SERCOS III ha esteso gli IDNs già presenti nelle precedenti versioni a una lunghezza di 32 bit andando a formare la seguente struttura:

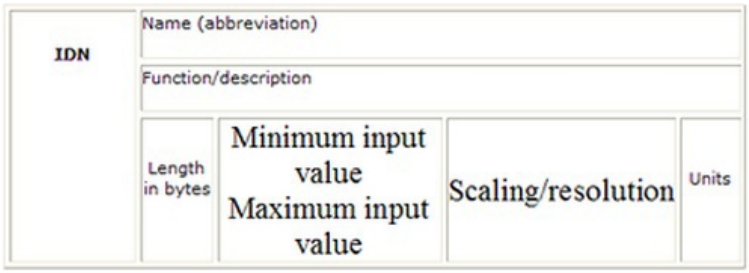


Fig. 3.13: Struttura IDNs

S-<DataSet >-<IdNr >.<SI >.<SE>

- La prima lettera rappresenta:
  S = standard IDNs
  P = IDNs specifici del dispositivo

- DataSet rappresenta il numero del set di dati (attualmente solo il set 0 è supportato)
- IdNr rappresenta il numero di identità SERCOS del comando, in ogni caso un numero decimale di 4 cifre
- SI rappresenta l'istanza della struttura. In una connessione tra master e slave c'è almeno un produttore e un consumatore di dati, la connessione produttore si crea tra il master e lo slave, mentre la connessione consumatore si crea tra lo slave e il master. Questa distinzione di connessioni è rappresentata all'interno del parametro SI.
- SE rappresenta l'istanza dell'elemento.

Tramite questi IDNs il master può parametrizzare l'impianto o comandare una funzione specifica inserendo nel pacchetto diretto agli slave il codice identificativo adatto. Questo permette di avere una struttura della comunicazione ordinata e facilmente leggibile e interpretabile.
Se ad esempio il master desidera comandare ciclicamente la posizione agli slave e leggere ciclicamente la coppia dei motori, sarà sufficiente configurare il MDT in modo che contenga S-0-0047 e il AT che contenga S-0-0084. Il gruppo degli IDNs standard (S-x-yyyy) è formato da 32'767 numeri, oltre 500 di questi sono usati per definire una serie completa di comandi indirizzati al motion control e alla gestioni degli ingressi e uscite (I/O commands). Tutti i prodotti SERCOS includono un sottoinsieme di questi IDNs utili allo svolgimento del proprio scopo, e possono non includere altri IDNs creati per altri scopi.
Al fine di non limitare lo sviluppo dei prodotti SERCOS, lo standard IEC permette ai produttori di aggiungere agli identificatori standard altri 32'767 IDNs proprietari cioè progettati dal produttore del dispositivo per svolgere particolari funzioni in aggiunta a quelle standard. Questi IDNs indicati con la lettera P (P-x-yyyy) vengono creati solamente per eseguire funzioni che non sono già coperte da un identificatore standard, questo per garantire interoperabilità tra dispositivi dello stesso tipo.

3.9 Diagnostica dell'impianto e sicurezza

SERCOS III offre una serie di opzioni diagnostiche per monitorare il corretto funzionamento della rete e dei dispositivi. Lo Status/Control word è un esempio già incontrato di diagnostica persistente sulla rete, in quanto viene utilizzata ad ogni ciclo di comunicazione.

Se uno dei bit di questa diagnostica segnala un malfunzionamento, è necessario utilizzare una diagnostica più approfondita per indagare l'origine del guasto.
Esistono quindi tre parametri standard: S-0-0011, S-0-0012 che eseguono una diagnostica sui dispositivi e s-0-0014 che segnala problemi sulla rete SERCOS III.
L'identificativo S-0-0011 esegue una diagnostica di classe 1 e viene utilizzato quando nella status word manda il bit 13 viene settato a 1. In risposta a questa richiesta di diagnosi vengono spediti in rete due byte con il seguente significato:

Bit no.	Errore
15	definito dall'utente
14	riservato
13	superato limite di posizione
12	errore di comunicazione
11	eccessiva deviazione del controllo
10	errore di fase nell'alimentazione
9	errore tensione inferiore al minimo consentito
8	errore tensione superiore al massimo consentito
7	errore corrente superiore al massimo consentito
6	errore in autocommutazione
5	errore nell'encoder
4	errore di controllo di tensione
3	raffreddamento spento
2	spegnimento per surriscaldamento motore
1	spegnimento per surriscaldamento driver
0	spegnimento per sovraccarico motore

Ognuno di questi errori causa lo spegnimento dello slave. L'identificativo S-0-0012 invece, viene segnalato dalla commutazione a 1 del bit 14 della status word e segnala un problema di classe 2, cioè un avviso di gravità inferiore rispetto agli errori.

Bit no.	Errore
15	Avviso specificato dal produttore
14	riservato
13	Posizione fuori dallo spazio di lavoro
12	Avviso di comunicazione
11	Deviazione di velocità eccessiva
10	Riservato
9	Alimentazione bus inferiore al minimo
8-6	Riservato
5	Riferimento di velocità superiore al massimo consentito
4	Riservato
3	Avviso sistema raffreddamento
2	Avviso surriscaldamento motore
1	Avviso surriscaldamento driver
0	Avviso sovraccarico motore

L'identificativo S-0-0014 si occupa di monitorare lo stato della rete e l'attuale fase di comunicazione.

Bit no.	Errore
15-14	Riservato
13	cambio di fase senza settaggio bit CPS (solo su SERCOS III)
12	cambio fase fuori tempo massimo (solo su SERCOS III)
11	errore di sincronizzazione IPO (solo su SERCOS II)
10	Drive con lo stesso indirizzo bloccati (solo su SERCOS II)
9	Transizione in uno stato non inizializzato
8	Cambio di fase senza il messaggio di conferma
7	Errore in fase di retrocessione (non in fase 0)
6	Errore nella sequenza delle fasi (sequenza non valida)
5	Fase non valida (fase <4)
4	Errore in MDT (solo su SERCOS III)
3	Errore in MST (superato il limite imposto con s-0-1003)
2-0	Fase di comunicazione: 000 = fase 0 001 = fase 1 010 = fase 2 011 = fase 3 100 = fase 4 101 = NRT

Oltre a questi identificativi, ogni driver esegue il monitoraggio di tutti i MDT e MST che riceve controllando:

- Corretto tempo di ricezione
- Lunghezza telegramma coerente
- Il codice CRC

Ogni infrazione del tempo di ricezione incrementa un contattore di errori: S-0-1028.0.0, fino a un massimo di 65535 . Mentre ad ogni infrazione riguardante la lunghezza del telegramma o il CRC si incrementa S-0-1035.0.0 con le stesse modalita. Una volta raggiunto questo limite oppure un limite definito dall'utente attraverso l'identificativo S-0-1003 il driver va in blocco e segnala il malfunzionamento con una comunicazione di errore di classe 1.

CIP safetyTM è un protocollo creato da ODVA (Open DeviceNet Vendors Association) che permette trasmissioni sicure attraverso i dispositivi presenti sulla rete. Certificato e già implementato in altre reti di comunicazione industrial ethernet come DeviceNet e EtherNet/IP è un protocollo accettato dal mercato mondiale, questo fatto è una garanzia sulla sua futura evoluzione e una riduzione dei costi per i componenti creati.
CIP Safety su SERCOS è una specifica funzione-profilo (FSP Safety) utilizzata per trasmettere dati relativi alla sicurezza su SERCOS III garantendo il livello di protezione SIL 3 (Safety Integrity Level) in accordo con IEC 61508.
SIL 3 corrisponde a un livello di probabilita di malfunzionamento on demand compresa tra 0.001 e 0.0001 e un fattore di riduzione del rischio pari a 1000-10000.
Per ottenere tutto questo non è richiesto l'utilizzo di un bus aggiuntivo, le informazioni sulla sicurezza sono spedite in aggiunta al traffico normale di SERCOS. La convergenza di drive, I/O, periferiche, bus-safety e standard Ethernet in un'unica rete semplifica la gestione e riduce i costi

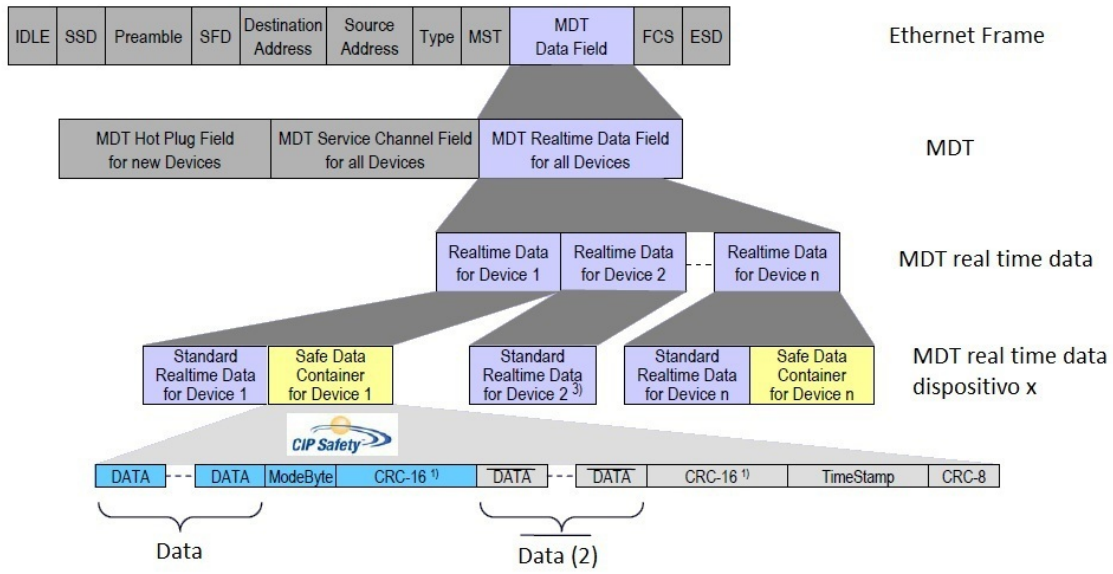


Fig. 3.14: CIP Safety su SERCOS III

di hardware e di installazione, in pi‘u rende facile da implementare controllori di sicurezza integrati e soluzioni di sicurezza omogenee.
Come si può osservare in figure 3.14 con CIP Safety su SERCOS III i dati riguardanti la sicurezza vengono inviati tramite lo stesso mezzo con lo stesso standard di comunicazione, in questo modo sia dispositivi standard che i dispositivi safety possono operare simultaneamente nella stessa rete. Il protocollo lavora a tutti i livelli della rete, a partire dalla comunicazione tra slave appartenenti allo stesso master fino ad arrivare a alla comunicazione tra dispositivi SERCOS III su reti diverse, e grazie alla capacità di routing del protocollo CIP è possibile comunicare dati riguardanti la sicurezza con altre reti non SERCOS III che utilizzano CIP safety. Per eseguire correttamente questo protocollo non è necessario che il master sia anche il controllore della sicurezza, in quanto è possibile instradare i dati senza essere in grado di interpretarli. Questo rende possibile per gli utenti configurare in modo flessibile l’architettura di rete per l’attuazione dei controllori di sicurezza programmabili o nella comunicazione tra sensori e attuatori.
Il protocollo safety permette di scambiare dati di dimensioni comprese tra 2 e 250 byte, per fare questo SERCOS III usa il servizio Sercos Messaging Protocol (SMP), che permette la frammentazione nei data container dei dati CIP ciclici. Attraverso l’identificativo S-0-1100.x.y è possibile configurare n contenitori di dati di lunghezza variabile all’interno dei telegrammi MDT/AT.
Associando la sicurezza del protocollo SERCOS Safety con la tolleranza degli errori fornita dalla ridondanza hardware di SERCOS III soluzioni di automazione altamente affidabili possono essere implementati per applicazioni di sicurezza, sia centralizzate che decentralizzate.

3.10 Performance e ultimi sviluppi di SERCOS III

Come ultima analisi osserviamo il comportamento della rete in condizioni di traffico Real Time (RT) e non Real Time (NRT) crescente:

Dati ciclici	tempo di ciclo	N° di slave (1)	N° di slave (2)	N° di slave (3)	N° di MDT/AT
8 byte	31.25 μs	7		2	1/1
12 byte	62.50 μs	14		8	1/1
16 byte	125 μs	26		21	1/1
12 byte	250 μs	61	30	57	1/1
32 byte	250 μs	33	17	31	1/1
12 byte	500 μs	122	94	120	2/2
50 byte	1 ms	97	85	95	4/4
32 byte	1 ms	137	120	134	4/4
12 byte	1 ms	251	220	245	4/4

(1) senza canale NRT
(2) con canale NRT 1500 bytes = 125 μs
(3) con canale NRT 250 Bytes = 20 μs

Questi dati sono stati forniti da SERCOS international nel settembre 2008, si nota che con tempi di ciclo minori di 250 μs il canale IP è più corto della lunghezza del frame massimo ethernet e quindi il traffico IP viene diviso in pi‘u pacchetti.
Osserviamo inoltre che i tempi di ciclo sono molto bassi, questo rende SERCOS III una delle migliori applicazioni per quanto riguarda il Motion Control, dove la capacità di chiudere gli anelli di controllo in tempi brevi permette maggior precisione e sicurezza sul processo. Facendo una classifica su base velocistica delle reti SERCOS III si colloca sotto a EtherCAT che è mediamente 2,7 volte più veloce nei medesimi scenari di utilizzo e sopra a ProfiNET che garantisce un tempo di ciclo minimo di 250 μs.

Per quanto riguarda le novità, nel terzo trimestre 2011 SERCOS international ha rilasciato la versione 1.3 delle specifiche SERCOS III. I principali cambiamenti sono:

- Supporto per differenti tempi di ciclo dei produttori: Non è più necessario aggiornare tutti i nodi nella rete nello stesso tempo di ciclo.
- Introdotto una applicazione di marca temporale: alla risoluzione di 1 ms per creare un sistema su vasta scala temporale, non per la sincronizzazione.
- Meccanismo di 3 buffer per l’analisi dei dati.
- Introdotto il tipo di dato sovra-campionato.
- Mantenuta la retrocompatibilità con le versioni precedenti (1.1.2).

Questa versione è ancora in fase di test e non è considerata stabile.

Capitolo 4

EtherNet/IP

4.1 Breve descrizione storica

Come per SERCOS III si inizia l'analisi della rete ripercorrendo le tappe fondamentali della nascita e sviluppo della rete. Per prima cosa è bene tenere presente che EtherNet/IP non nasce come rete ex-novo senza un predecessore storico nei bus di campo, ma si può considerare come l'evoluzione in chiave ethernet delle reti DeviceNet e ControlNet prodotte sempre da Allen Bradley ed implementa sempre il protocollo Common Industrial Protocol (CIP^{TM}).
La nascita di Ethernet/IP può essere collocata alla fine del 1990, quando Allen-Bradley, azienda successivamente acquisita da Rockwell Automation che diffuse le prime pubblicazioni sulle specifiche della trasmissioni di messaggi e del trasferimento Real-Time dei dati, cioè rispettivamente 1998 e 1999. Questo lavoro era parte di un progetto di più ampio respiro: la formulazione delle specifiche Control Net International (CI).
Nel 2000 l'associazione Open DeviceNet Vendor Association (ODVA) che sostiene le reti basate su CIP, affianca Allen Bradley nella gestione e sviluppo della rete, mette on-line il toolkit di EtherNet/IP e organizza il primo corso di formazione per utenti di questa nuova rete.
Successivamente a ODVA anche altre associazioni come l'organizzazione per la promozione di ethernet su piattaforme indipendenti IAONA , e l'organizzazione per lo sviluppo del protocollo "IDA "confermano il proprio supporto e creano il "Memorandum of Understanding "volto ad unire le forze per lo sviluppo della rete EtherNet/IP.
Nel 2001 EtherNet/IP riceve il primo riconoscimento ufficiale, l'Editors' Choice Award 2000 da parte del comitato Control Engineering. Nello stesso anno vengono pubblicate sui siti delle associazioni collaboratrici: www.odva.org, www.ethernetip.org le specifiche della rete.
Dal punto di vista hardware, nell mercato mondiale molte aziende hanno dato la loro fiducia a questa nuova rete e già dal 2000 oltre 80 compagnie producevano e immettevano nel mercato materiale specifico per lo standard EtherNet/IP, offrendo all'utenza una ampia rete di venditori.
Nel settembre del 2001 si è svolto a Detroit il primo workshop dedicato alle implementazioni EtherNet/IP, un incontro di ampio respiro dove è sono stati valutati i progressi tecnologici della rete e studiate le direzioni da prendere per rendere più competitiva la rete. L'anno successivo si è aperto un work-group collaborativo tra ODVA IDA e PNO sempre finalizzato a migliorare le caratteristiche della rete.
Lo stesso anno il TÜV (Technischer Überwachungs-Verein, Associazione di Controllo Tecnico, organo di certificazione tedesco in ambito di sistemi di gestione sicurezza alimentare e ambientale e per la qualità del sistema di gestione aziendale) approva le specifiche CIPSafety che verranno rilasciate da ODVA l'anno successivo.
Nel 2003 il protocollo CIP viene riconosciuto come Standard IEC/EN 61158 e 6784-1. Nel 2003 e 2004 vengono rilasciate 2 specifiche fondamentali per la rete, CIP Sync e CIP Motion che assieme alla già pubblicata CIP safety permette a EtherNet/IP di entrare a pieno titolo nel modo delle reti

specifiche per Motion Control estendendo il suo campo di impiego anche nei processi automatici tempo critici.
Nel 2006 vengono immessi nel mercato i primi dispositivi CIP Safety per EtherNet/IP e successivamente i prodotti con specifiche CIP Sync e CIP Motion, offrendo attualmente sul mercato un'ampia offerta di prodotti adatti alle esigenza dell'industria moderna.
Nel futuro dell'azienda c'è la produzione di componentistica particolare per migliorare il supporto alla specifica CIP Motion e la ricerca e sviluppo di tecnologie per portare la comunicazione wireless in ambito industriale.

4.2 EtherNet/IP hardware

Ethernet/IP a differenza di SERCOS III utilizza lo standard ethernet non modificato, senza cioè apportare modifiche ai protocolli standard. Questo permette di utilizzare tutto l'hardware in commercio progettato per lavorare con TCP/IP e UDP/IP riducendo i costi e garantendo una disponibilità e un'interoperabilità massima tra i dispositivi.
A livello di comunicazione sono supportati gli standard di comunicazione ethernet su cavo in rame CAT5e e su fibra ottica, alle velocità di 10 Mbps 100 Mbps e 1 Gbps. A livello topologico è possibile costruire la propria rete in tutti i modi ammessi dallo standard IEEE802.3, ossia utilizzando le forme già viste ad anello o lineare, applicando il concetto di mezzo fisico condiviso o tipo Trunked. Oppure sfruttando le infrastrutture tipiche dalla tecnologia switching ethernet e si possono ottenere reti a stella, ad albero o a switch-ring creando soluzioni bilanciate tra ridondanza nella comunicazione e numero di connessioni da effettuare.

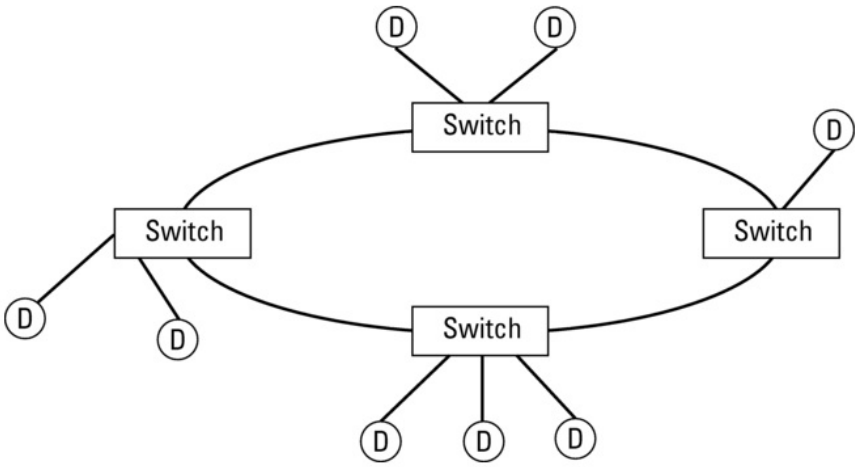


Fig. 4.1: Esempio topologia switch ring

É bene specificare che anche se tutto l'hardware standard ethernet è compatibile con EtherNet/IP, nel senso che funziona, il livello di prestazioni raggiungibile può variare notevolmente a seconda del modello scelto. Switch, router, e in generale tutti i dispositivi presenti nell'infrastruttura della rete introducono ritardi dovuti al proprio meccanismo di funzionamento interno; questa variabilità di prestazioni ha storicamente escluso EtherNet/IP dal cerchio delle reti per Motion Control, relegandola ad ambienti diversi, dove un comportamento deterministico poco spinto e jitter elevati non erano un problema.
Recentemente si è assistito a un cambio di tendenza in questo ambito, le specifiche di CIP Sync e CIP Motion basate sullo standard IEEE 1588, hanno introdotto nuove caratteristiche per l'hardware EtherNet/IP. In particolare sono stati modificati gli switch e i router in modo tale da supportare questa tecnologia. In questo modo da un lato è venuto meno il concetto di fare Real Time

con tutto l'hardware ethernet disponibile, dall'altra con la creazione di nuovi standard sono stati ottenuti progressi concreti nelle performance della rete.
Ed è proprio grazie a questi progressi che oggi la rete EtherNet/IP con CIP Sync e CIP Motion, è diventata una scelta valida in campo di Motion Control e comunicazioni Real Time.

4.3 EtherNet/IP logica di funzionamento

EtherNet/IP garantisce comportamenti deterministici in maniera completamente diversa da SERCOS III, in quanto non utilizza il master per gestire la comunicazioni su un mezzo condiviso, ma le infrastrutture tipiche del lan-switching, in modo da eliminare il concetto stesso di mezzo condiviso.
Questo avviene praticamente utilizzando dei dispositivi intelligenti chiamati switch che effettuano le connessioni tra i nodi della rete. Attraverso di essi si crea un canale di comunicazione full-duplex tra ogni coppia di porte dello switch, in questo modo le collisioni non possono avvenire perchè ogni comunicazione avviene su un mezzo trasmissivo distinto degli altri e riservato.
Gli switch moderni offrono inoltre altre funzioni utili:

- Applicano il modello store and forward. Lo switch attende la completa ricezione del messaggio prima di inoltrarlo al destinatario; in questo modo è possibile eseguire il controllo di ridondanza ciclica sul pacchetto per segnalare eventuali errori, riducendo drasticamente il traffico di pacchetti errati.
- Implementano la funzione di Internet Group Management Protocol (IGMP). Sono in grado cioè di creare e gestire delle liste di nodi in maniera tale da non inondare la rete con i messaggi multicasting, ma di inoltrarli solo ai nodi inseriti nella lista multicast, questa procedura viene chiamata IGMP snooping.
- Priorità dei messaggi. Viene inserito in tutti i messaggi un valore che ne descrive la sua priorità; in questo modo lo switch che riceve più messaggi diretti allo stesso destinatario può decidere l'ordine con cui spedirli in rete garantendo una certa qualità e classe del servizio.
- Virtual Local Area Network (VLAN). Attraverso una specifica funzione la rete viene suddivisa in gruppi di nodi a formare una lan con il proprio dominio di collisione; in questo modo il traffico di pacchetti generati in una VLAN non circola in quelle vicine rendendo il traffico più ordinato e circoscritto.

Ma questa tecnologia ha un pesante limite: la congestione interna allo switch dei messaggi che condividono lo stesso destinatario.
Nel semplice caso in cui ogni mittente spedisce il proprio messaggio a un destinatario diverso, lo switch svolge il proprio compito come sopra. Ma nel caso tipico delle reti industriali dove più nodi (slave) spediscono il proprio messaggio a un solo nodo (Master), ci si scontra con il limite fisico dello switch: può essere spedito infatti un solo messaggio alla volta, mentre gli altri vengono messi in coda e spediti di seguito. Questo ritardo costituisce il maggior limite alle performance real-time della rete EtherNet/IP, e anche se mitigato attraverso i meccanismi sopra descritti, sarà sempre presente.

Dal punto di vista del protocollo EtherNet/IP è una rete di comunicazione orientata alla connessione e basa il suo funzionamento sullo standard CIP. É necessario infatti creare una connessione tra i dispositivi prima di iniziare a trasferire dati tra loro; proprio il protocollo CIP che verrà analizzato nel prossimo paragrafo definisce la struttura di questa connessione, l'organizzazione

dei messaggi e dei dati che circolano in rete.

4.4 Il protocollo CIP

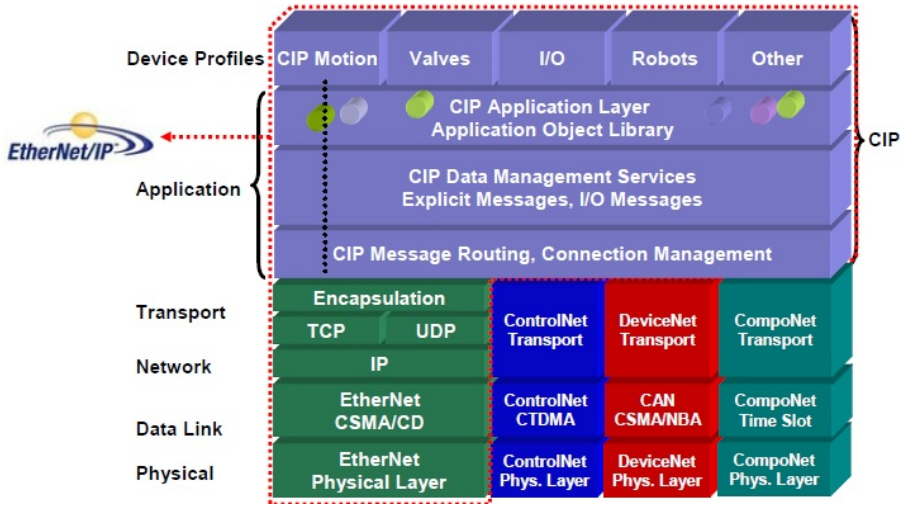


Fig. 4.2: Protocollo CIP

Il protocollo Common Industrial Protocol (CIP) risiede al livello applicazione del modelli ISO/OSI e svolge le funzioni di indirizzamento messaggi, gestione delle connessioni, definizione degli oggetti, oltre alle funzioni introdotte dai pacchetti aggiuntivi CIP Sync e Motion.
L'idea alla base di questo protocollo è il modello produttore consumatore, dove ogni nodo che genera dei dati come un sensore di temperatura o un encoder viene definito il produttore della risorsa e tutti i nodi che desiderano conoscerla sono definiti consumatori.
Questo modo di vedere la rete permette di semplificare le comunicazioni multicast ovvero tutte le trasmissioni da un unico nodo produttore verso molti nodi consumatori, questa funzionalità è ottenuta praticamente identificano i messaggi non più attraverso il loro indirizzo di destinazione, ma bensi attraverso un identificativo di connessione (CID). In questo modo il produttore di risorse spedisce in rete i propri dati contrassegnandoli in maniera univoca e comprensibile a tutti, mentre i nodi consumatori che controllano il traffico in rete analizzano solo il campo identificativo dei messaggi, andando cosi a prelevare e consumare solo i messaggi desiderati.

Il protocollo CIP per trasmettere le informazioni in rete definisce due tipi di connessioni/messaggi : Messaggio esplicito e messaggio implicito.

Messaggio implicito	Messaggio esplicito
Offre il servizio di comunicazione connessa	Offre entrambi i servizi di comunicazione connessa e non connessa
Protocollo di comunicazione UDP/IP	Protocollo di comunicazione TCP/IP
Utilizzato principalmente per comunicazione di I/O ciclici e dati Real-Time	Utilizzato per comunicazioni di parametrizzazione e diagnostica
Creato dall'oggetto producing application e consumato da uno o più consuming application	creato dall'oggetto Message-Router
Ogni messaggio viene definito implicitamente dal ID connessione	ogni messaggio contiene le informazioni specifiche riguardanti il nodo destinazione

Tra i due quello utilizzato durante le comunicazioni real time della rete è quello implicito. Nello specifico è possibile differenziare questo messaggio in 4 tipologie diverse, ciascuna studiata e utilizzata per una particolare funzione:

-polled:
Si tratta del classico messaggio con cui il master istruisce sequenzialmente tutti gli slave sul compito da svolgere e riceve in risposta i dati dal campo.

-strobed:
In questo caso il master spedisce un'unico messaggio in multicasting a tutti gli slave, e riceve sequenzialmente risposta da loro

-ciclico:
I messaggi ciclici sono prodotti da un dispositivo sulla base di una schedulazione predeterminata e vengono messo in rete identificati da un particolare ID in modo tale che tutti i consumatori possano accedervi. Sono i messaggi di gran lunga più utilizzati nella pratica industriale

-cambio di stato:
Molto simile al messaggio ciclico, ma viene trasmesso in rete in seguito ad un evento che accade nell'impianto quindi slegato da una programmazione temporale. Viene in genere spedito regolarmente in rete con una certa frequenza per prevenire un suo eventuale malfunzionamento.

Il protocollo CIP suddivide i dispositivi hardware in tre classi, in base al tipo di traffico che generano sulla rete:

- Classe Messaging:
  I dispositivi appartenenti a questa classe sono in grado di supportare i messaggi espliciti (connessi e non) prodotti dalle altre classi di dispositivi, ma non possono ricevere o trasmettere dati real-time.
  I più comuni dispositivi appartenenti a questa classe sono i PC utilizzati per programmare l'impianto, i dispositivi di interfaccia uomo-macchina, tools di diagnostica e configurazione della rete etc.

- Classe Adapter:
  Sono i consumatori dei dati Real-Time prodotti dai dispositivi appartenenti alla tipologia Scanner. Questi dispositivi non sono in grado di trasmettere o ricevere dati real-Time senza che un dispositivo scanner inoltri la richiesta e inoltre non salvano e non generano i dati necessari per stabilire la comunicazione.
  Questa classe quindi scambia messaggi di tipo esplicito sulla rete con gli altri dispositivi ma non può originare la comunicazione. Alcuni esempi di questi dispositivi sono: Driver di motori, Robot, Rack di I/O real-Time etc.

- Classe Scanner:
  Questi prodotti infine generano la richiesta di connessione esplicita e dialogano con tutti gli altri dispositivi presenti sulla rete. L'esempi classico di questi dispositivi è il PLC, o il controller della rete, i Master etc.

Un'altra caratteristica del modello di comunicazione EtherNet/IP è di essere fortemente orientato agli oggetti, lo standard CIP modellizza ogni dispositivo fisico come un insieme di oggetti a di interazione tra di essi. A tale scopo definisce 46 classi di oggetti e solo la minoranza di essi sono legati al particolare tipo di rete utilizzata (ControlNet, DeviceNet, EtherNet/IP), la maggioranza

sono di tipo comune utilizzabili in tutte e tre le reti.
Gli oggetti sono classificati in base al tipo di dispositivo a cui vengono applicati, insiemi di oggetti formano degli standard operativi comuni in tutti i dispositivi appartenenti a una certa classe, semplificano la manutenzione dell'impianto e l'interoperabilità tra i dispositivi.

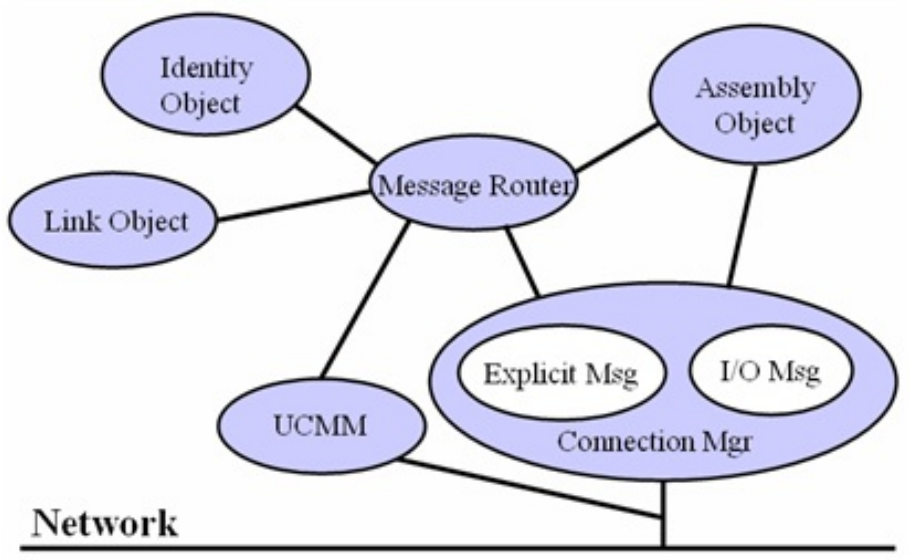


Fig. 4.3: Struttura degli oggetti in CIP

Per conoscere quali oggetti sono contenuti nei vari dispositivi, ogni produttore crea un EDS (Electronic data sheet) del proprio prodotto che ne descrive il funzionamento e le caratteristiche.
In particolare esistono tre oggetti standard che ogni dispositivo contiene e utilizza per svolgere le proprie funzioni:

- Oggetto richiesta:
  Questo oggetto è obbligatorio in ogni dispositivo CIP e include l'oggetto identità, l'oggetto Message-Router, e l'oggetto specificativo della rete.

- Oggetto Applicazione:
  Dipende fortemente dal tipo di dispositivo e dalle sue funzioni, questo oggetto definisce come il dispositivo organizza i dati per la trasmissione in rete.

- Oggetto definito dal produttore:
  In questo oggetto sono contenuti tutti gli oggetti non standard definiti dal produttore per caratterizzare il funzionamento e le peculiarità del proprio dispositivo.

Alcuni esempi di oggetti comuni sono:

-Oggetto Identità: Classe ID 0x01, è un oggetto di sola lettura tranne che per un campo e definisce le informazioni primarie del dispositivo secondo la seguente tabella.

Attributi obbligatori	Attributi opzionali
ID produttore	Stato
Tipo di dispositivo	Valore di consistenza configurazione
Codice prodotto	Intervallo di frequenza
Revisione	
Status	
Numero Seriale	
Nome prodotto	

-Oggetto Message-Router: si fa carico della distribuzione dei messaggi espliciti ai relativi oggetti applicazione.

Più istanze dello stesso oggetto possono coesistere in un dispositivo in questo caso ci si riferisce ad esse con il nome di classi.

La struttura ad oggetti tipica di CIP viene utilizzata anche per gestire la comunicazione in EtherNet/IP. Attraverso l'utilizzo di oggetti specifici come schematizzato in figura 4.4 viene instaurata la connessione tra due dispositivi, scambiati messaggi e terminata la connessione.

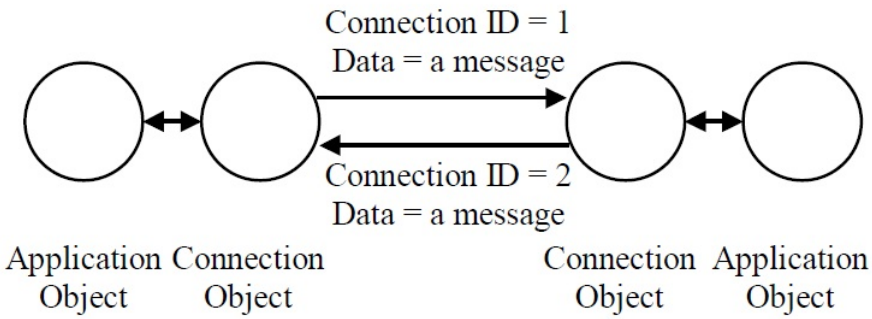


Fig. 4.4: Connessione e ID connessione

Per quanto riguarda la gestione dei dati all'interno del dispositivo, gli oggetti Assembly si occupano di contattare gli oggetti applicazione, accedere ai loro dati, e con essi costruire il messaggio che successivamente sarà spedito.
Questi messaggi sono strutturati attraverso il sistema di indirizzamento CIP Segment cosi formato:

- Device network address: Può essere l'indirizzo del nodo della rete o un identificativo MAC (Medium Access Control)
- Class ID: Questo campo contiene l'identificativo della classe a cui appartiene l'istanza
- Istance ID: l'istanza stessa
- Attribute ID: contiene gli attributi di interesse
- Service Code: Descrive le azioni e i servizi necessari a soddisfare la richiesta

Ogni parte che compone il CIP segment può avere dimensioni diverse, 1 Byte, 2Byte o 4 Byte.

4.5 CIP Sync, CIP Motion e CIP Safety

L' EtherNet/IP classico, come già anticipato, non garantisce trasmissioni deterministiche di alto livello come quelle richieste da applicazioni di Motion Control. E per risolvere questa mancanza ODVA, IOANA, IDE hanno creato due integrazioni al protocollo CIP: CIP Sync e CIP Motion.

CIP Sync

Questo protocollo nasce dall'esigenza di fornire una sincronizzazione dei tempi tra i vari componenti dell'impianto. Basato sullo standard IEEE-1588 (PTP -Precision Time Protocol) fornisce le regole per una precisa sincronizzazione del clock di tutti i dispositivi presenti nell'impianto, anche se dotati di precisione e risoluzione differenti tra loro. Il livello di risoluzione temporale raggiungibile con questa tecnologia è dell'ordine del ηs ed è possibile sincronizzare il clock nei dispositivi con una tolleranza di $\pm 100\eta s$.
Permette inoltre il trasferimento di informazioni ad intervalli di tempo prestabiliti con il minimo jitter possibile in modo tale da garantire il sincronismo necessario alla pratica industriale.

Il meccanismo con cui avviene la sincronizzazione dei tempi si basa su una serie di misure eseguite tra i dispositivi collegati all'impianto.
Per prima cosa in fase di inizializzazione della rete si svolge una "gara" dove ciascun dispositivo comunica la classe del proprio oscillatore locale; da questo confronto esce vincitore quello che garantisce il clock più preciso. Se nessuno dei dispositivi utilizzati presenta un classe adeguata alle esigenze è possibile collegare all'impianto un ricevitore GPS ottenendo quindi la massima precisione possibile. Il vincitore diventerà il Master Clock, cioè il dispositivo che fornirà la base temporale comune a tutto l'impianto e si occuperà di sincronizzare gli dispositivi presenti.
Una volta scelto il Master Clock, questo svolgerà delle misure come schematizzato in figura 4.5, al fine di stimare la deviazione dei clock locali dal Master Clock. Si analizza ora i passi di questa procedura:

1. Il master Clock spedisce in rete un sync-message con all'interno il time-stamp dell'istante previsto di invio.

2. Successivamente, sempre dal Master Clock, invia un nuovo messaggio contenente il tempo in cui il sync-message è stato effettivamente spedito.

3. Con queste informazioni ogni slave presente nella rete calcola la deviazione del proprio clock da quello del master e effettua localmente la sincronizzazione.

Questa procedura trascura però ancora un problema, la compensazione del ritardo di trasmissione della rete. Questo valore è infatti variabile a seconda di dove si trovi il dispositivo sulla rete e che infrastrutture deve attraversare il pacchetto per raggiungerlo.
attraverso un ulteriore scambio di messaggi tra master e ciascun slave è possibile calcolare e compensare anche questo ritardo.

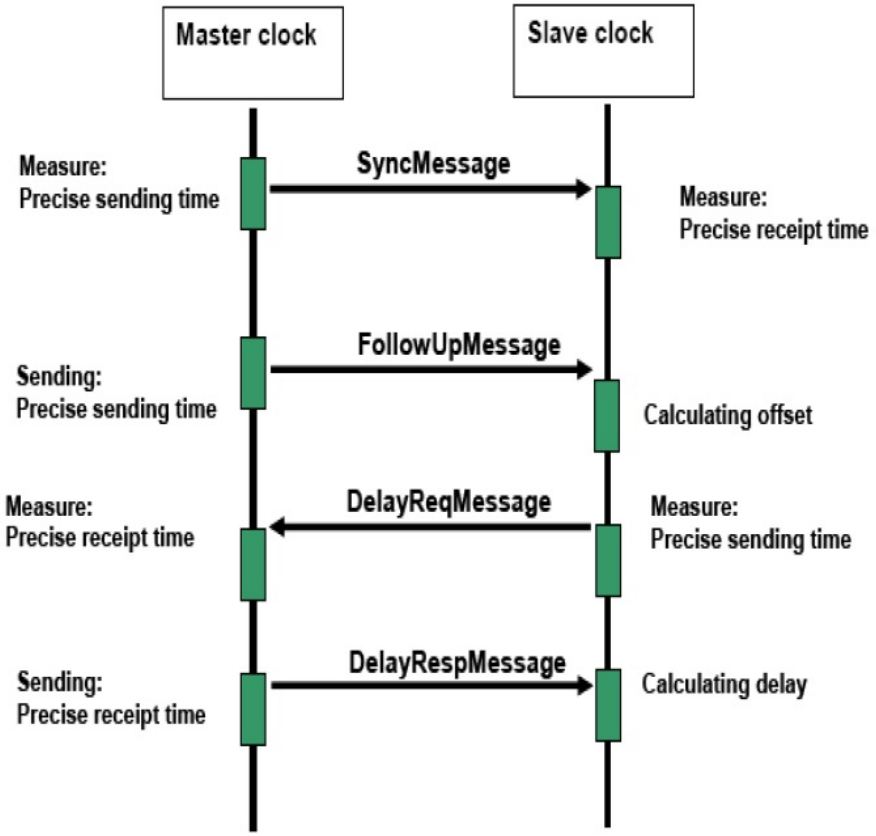


Fig. 4.5: Meccanismo di CIP Sync

- Ciascun slave spedisce un messaggio di tipo delay-req-msg con il time stamp di spedizione, ovvero l'informazione temporale dell'istante di invio.
- Il master risponde con il delay-resp-msg, un messaggio che contiene il time stamp di arrivo del messaggio delay-req-msg.
- Semplicemente osservando lo scarto temporale tra trasmissione e ricezione è possibile calcolare e compensare anche questo ritardo.

In questa trattazione si è parlato del time-stamp, un marchio temporale che indica il momento esatto in cui i messaggi vengono ricevuti e spediti dal dispositivo. É essenziale ai fini della precisione che questo riferimento temporale venga generato al livello più basso possibile nel modello ISO-OSI per non introdurre ritardi dovuti alla gestione dei frame del pacchetto quando transita verso i livelli alti del modello.

Grazie a questo è possibile ottenere un comportamento deterministico dell'impianto con una comunicazione non perfettamente deterministica (a causa dei ritardi introdotti dagli switch). L'aver inserito l'informazione temporale nel pacchetto permette al dispositivo ricevente di legare l'informazione acquisita a un istante ben definito del ciclo industriale e agire di conseguenza.

CIP Motion

Cip Motion è l'applicazione dalla sincronizzazione temporale introdotta da CIP Sync nelle procedure di controllo assi distribuito tipici del Motion Control.
Attraverso l'inserimento in ogni messaggio dell'informazione temporale diventa possibile legare in maniera precisa gli spostamenti all'istante temporale, e applicare quindi i comuni sistemi di controllo.
Questa estensione contiene a tale scopo numerosi profili applicativi dedicati ai controlli di posizione, coppia e velocità all'interno dell'azionamento. Attraverso questo modulo è possibile collegare più azionamenti e creare sistemi multiasse con controlli di tipo camma elettronica o gearing.
Tutto questo è stato possibile senza modificare la struttura di ethernet, ma solo aggiungendo il time-stamp ai messaggi spediti e delegando ai dispositivi finali la gestione delle informazioni di temporizzazione necessarie per rispondere alle esigenze del controllo in tempo reale.
Il profilo CIP Motion fornisce inoltre i servizi di configurazione, stato e diagnostica dei dispositivi e crea un supporto comune per le istruzioni utilizzate negli applicativi, rendendo cosi slegata la programmazione della movimentazione dall'hardware in uso. Utilizzando inoltre i dati a virgola mobile si semplifica notevolmente la complessita matematica legata alle formule di motion control.
I risultati ottenuti nel Motion Control sono:
-coordinare 100 assi con un tempo di aggiornamento di rete di 1 ms.
-Ottenere intervalli di trasmissione costante con jitter minimo 100 *etas*.

Purtroppo sono risultati ottenibili solo in reti molto piccole, isolate e molto ottimizzate con una accurata scelta dei componenti.
Negli scenari industriali tipici le prestazioni decadono drasticamente, portando i produttori stessi della rete a consigliare altre standard come ControlNet e DeviceNet che offrono prestazioni migliori a un prezzo inferiore.

CIP Safety

CIP Safety è nata come rete di sicurezza al servizio del bus di campo DeviceNet ed è poi migrata su EtherNet/IP e altre reti di comunicazione real time ethernet migliorando le proprie caratteristiche grazie al throughtput maggiore offerto da ethernet rispetto ai bus.
A differenza degli altri approcci industriali, CIP Safety è stato sviluppato per essere indipendente dalla rete e garantendo il funzionamento sulla maggior parte dei sistemi di comunicazione industriali attualmente in commercio.
Il cuore di questo protocollo risiede nella grande quantità di misure operate sulla rete, volte non tanto ad impedire che un errore avvenga, cosa fra l'altro impossibile nel mondo industriale, ma ad effettuare una rilevazione accurata di ogni anomalia presente sui pacchetti, quali corruzioni di bit, ritardi, ripetizioni, etc.
I metodi di controllo adottati sono:
-Il Safety validator: si tratta di un oggetto presente in tutti i dispositivi CIP safety che svolge la funzione di creare i collegamenti e validare i pacchetti safety in entrata e uscita.
-Misure temporali tramite Time Stamp dei pacchetti. Permettono ai consumatori di conoscere l'età del pacchetto anche se questo attraversa dispositivi evoluti con ritardi intrinseci come i router o i bridge. Queste permette di rivelare eventuali ritardi eccessivi accumulati dalla rete.
-Controllo degli identificatori. Controllando i campi di mittente e destinatario o il codice identificativo tipico del messaggio/connessione viene garantito che ogni messaggio arrivi al destinatario corretto.

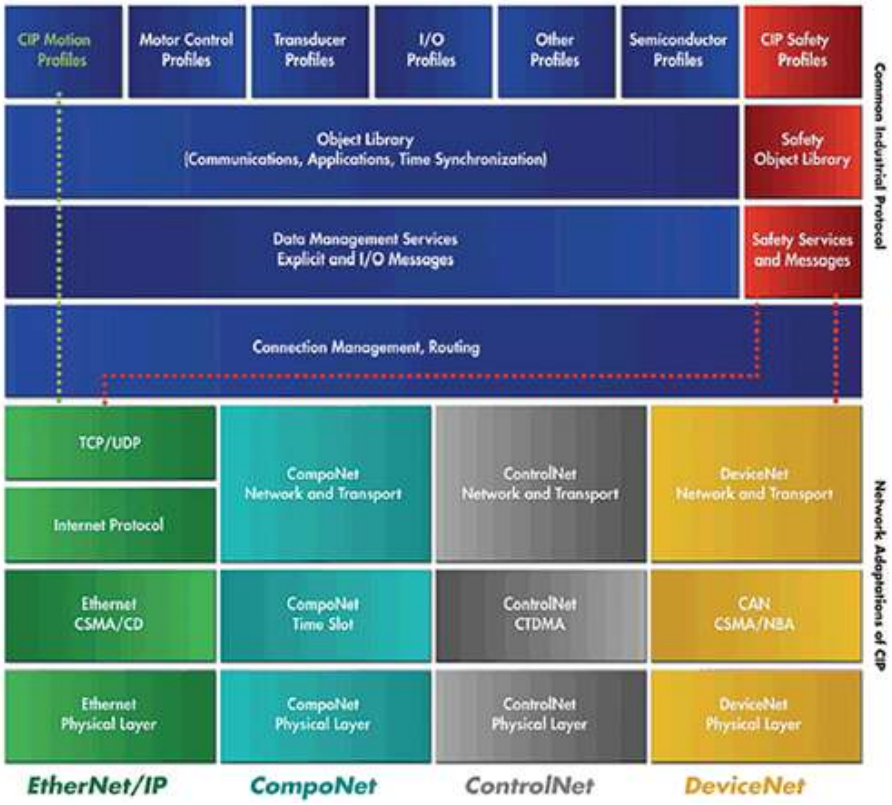


Fig. 4.6: Protocollo CIP Safety

-Controllo dei codici di ridondanza ciclica (CRC). Attraverso questo meccanismo viene monitorata l'integrità dei dati da parte dei consumatori del pacchetto rilevando e/o correggendo eventuali anomalie

La segnalazione di un errore da parte di questi controlli porta il dispositivo in uno stato particolare definito safe, le cui caratteristiche dipendono dal tipo di dispositivo, e rimane in questo stato finchè il Master non interagisce con esso.

La interoperatività di questo protocollo con reti di tipo differente può avvenire perchè la sicurezza della comunicazione viene affidata alla codifica del pacchetto e non agli strati del modello ISO/OSI sottostanti, quindi lo stesso pacchetto può partire su una rete come EtherNet/IP e terminare e terminare su di un'altra come DeviceNet che supporta CIP Safety.
Al fine di rendere snello il protocollo sono state definite due tipologie di pacchetti safety, una corta utilizzata per trasportare al massimo 2 Byte conn 8+8 bit di safety CRC, e una lunga per trasportare fino a 250 Byte con 16+16 bit di safety CRC. In questo modo si evita di congestionare la rete e si mantiene al contempo un bun livello di protezione dei dati.

4.6 Analisi frame EtherNet/IP

I pacchetti utilizzati nel funzionamento real time della rete EtherNet/IP sono i comuni pacchetti User Datagram Protocol la cui struttura è riportata in figura 4.7

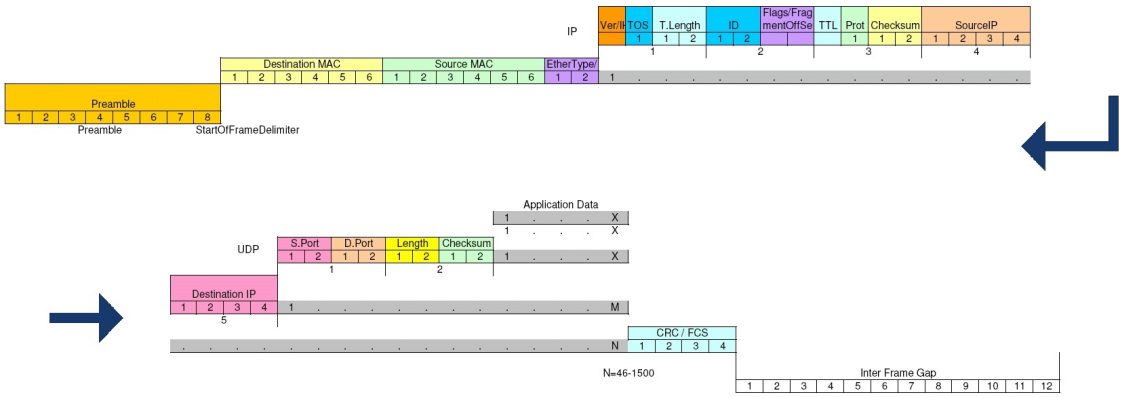


Fig. 4.7: Struttura pacchetto UDP

Analizziamo singolarmente i campi di cui è composta:

-Destination MAC: In questi 6 byte è contenuto l'indirizzo MAC del destinatario del messaggio.

-Surce MAC: nei 6 byte successivi è riportato l'indirizzo MAC del mittente del messaggio.

-EtherType: come In SERCOS III questo campo formato da 2 byte identifica il tipo di protocollo utilizzato.

-Ver: questi 4 bit rappresenta la versione IP utilizzata.

-TOS: Questo campo formato da altri 4 bit indica l'urgenza e il tipo di servizio che il messaggio svolge; attraverso questa informazione il messaggio ottiene una certa precedenza attraversando la rete. Non sempre è utilizzato, ma resta comunque presente nel messaggio.
Gli ultimi 5 bits indicano il tipo di servizio:
il bit di delay se settato a 1 indica la richiesta al router di utilizzare il percorso che introduce minor ritardo possibile.
il bit di throughput richiede al router di instradare il pacchetto sul percorso con throughput maggiore.
il bit di reliability richiede il percorso con minor probabilità di perdere il pacchetto.
Gli ultimi 2 bits sono riservati e posti sempre a 0.

-T-Length: questi 2 byte indicano la lunghezza dell'header riportando il numero di double word (1 word = 4 byte) di cui è composto. Il valore di default è 5 (20 byte di header). Numero superiori a 5 indicano delle opzioni presenti, mentre numeri inferiori indicano un errore.

-ID: formato da 2 byte rappresenta l'identificatore ID unico del datagramma. Se il datagramma viene suddiviso in più pacchetti tutti riportano lo stesso ID.

-Flag/fragmentOffset: Indica la lunghezza totale del pacchetto, compreso l'header e i dati contenuti.

-TTL: Time to Live byte, indica il numero di router o dispositivi di infrastruttura che il pacchetto può attraversare prima di essere scartato.

-PROT: Indica l'identificativo ID del protocollo utilizzato al livello superiore. In questo caso si utilizza User Datagram Protocol (UDP).

-Checksum: Questi 2 byte sono un campo di sicurezza chiamato header checksum che esegue un controllo sulla corretta ricezione del solo campo IP header.

-Source IP: questi 4 byte contengono l'indirizzo IP del mittente del messaggio

-Destination IP: questi 4 byte contengono l'indirizzo IP del destinatario del messaggio

-S Port: 2 byte contenenti l'indirizzo della porta del dispositivo mittente che ha generato il dato

-D Port:2 byte contenenti l'indirizzo della porta del dispositivo destinatario che deve ricevere il dato

-Length: 2 byte che indicano la lunghezza totale del messaggio inclusi i dati e l'header UDP.

-Checksum: questi 2 byte formano l'UDP checksum, un altro campo di sicurezza per garantire la corretta ricezione dei dati. É opzionale ma utilizzato in molte applicazioni.

-Data: questo campo di dimensione variabile da 46 a 1500 byte contiene i dati che si desidera trasportare.

-CRC/FCS: 4 byte di controllo di ridondanza ciclica sui dati trasmessi.

Nello specifico il campo data è formato dal protocollo CIP che può essere ulteriormente suddiviso nel seguente modo:

-Item Count: i primi due byte rappresentano il numero di common packet format a seguire, deve essere almeno 2. Nel protocollo UDP CIP questo valore sarà sempre impostato a 2.

-Type ID: 2 byte, rappresenta l'identificativo dei dati trasportati.

-Cip header length: 2 byte, indica la lunghezza del campo CIP header compreso il connection Identifier e il sequence number.

-Connection Identifier: questi 4 byte contengono il valore dell'ID caratteristico della connessione.

-Sequence Number: 4 byte, indicano la sequenza di pacchetti per questa particolare connessione.

-Data Type ID: Questi 4 byte indicano il tipo di dati utilizzato.

-Data length: Questi 4 byte indicano la lunghezza del campo dati.

-Data: di grandezza variabile in relazione all'applicazione e al dispositivo.

4.7 Performance e sviluppi futuri

Vengono riassunte nella seguente tabella dei risultati prestazionali della rete ordinati in base all'aumentare del numero di dispositivi connessi e all'aumentare delle prestazioni (e prezzo) del dispositivo/i switch utilizzato/i.
Il tempo di ciclo minimo (RPI-Requested Packet Interval) dipende dal numero di connessioni CIP, e ogni dispositivo può avere più di una connessione CIP. In questo caso si è calcolato il min-RPI = (numero di connessioni x 2)/(numero di frame/sec) assumendo che tutte le connessioni abbiano lo stesso tempo di scansione RPI.
In questo calcolo si sono trascurati i ritardi introdotti dagli switch, che Rockwell in mancanza di specifiche del produttore quantifica in 0.1 ms per dispositivo.

Numero di connessioni	Min-RPI (ms) con 5000 Frames/sec	Min-RPI (ms) con 10000 Frames/sec	Min-RPI (ms) con 25000 Frames/sec
4	1.6	0.8	0.32
8	3.2	1.6	0.64
16	6.4	3.2	1.28
32	12.8	6.4	2.56
64	25.6	12.8	5.12

Come si nota immediatamente, le prestazioni sono nettamente inferiori se confrontate con SERCOS III, e la stessa casa produttrice consiglia di orientarsi verso altre soluzioni come ControlNet se si è alla ricerca di prestazioni più spinte.
EtherNet/IP rimane comunque una rete valida per soluzioni integrate di soft real-time, ufficio , magazzino , contabilità etc.

Viste le prestazioni deludenti se confrontate con la concorrenza, ODVA nell'assemblea generale svoltasi nel 2009 ha annunciato importanti cambiamenti nel profilo CIP Motion per ridurre il gap prestazionale rispetto ad altre realtà:
-cambiare le procedure di inizializzazione della rete in modo da organizzare meglio la struttura dei dati e delle connessioni
-ridurre da 120 bytes a 36 bytes i dati di processo nel controllo drive in modo da snellire le comunicazioni
-aggiungere ai driver un "CIP Motion hardware assis" per decentralizzare il controllo del moto.

Proprio quest'ultimo punto, che richiederebbe l'introduzione di hardware specifico, in pratica un FPGA per gestire localmente il protocollo CIP, renderebbe EtherNet/IP una rete non standard ethernet in maniera simile a SERCOS III e etherCAT.
Questo fatto può essere interpretato come un'azione contraria alla filosofia di utilizzare solo hardware standard, ma ad oggi rappresenta l'unico modo per offrire agli utenti certe prestazioni.

Capitolo 5

Wireshark

Wireshark è un software multipiattaforma liberamente scaricabile dal sito:
http://www.wireshark.org
É stato scritto in linguaggio di programmazione C e distribuito al pubblico sotto la licenza GNU General Public License. Offre le funzioni di packet sniffer e packet analyzer e viene comunemente impiegato per diagnostica sulle reti di comunicazione, sviluppare protocolli e per scopi didattici.
Sviluppato nel 1990 da Gerald Combs e rilasciato al pubblico nel 1998 sotto il nome di Ethereal ha subito un lungo processo di evoluzione con oltre 600 collaboratori impegnati a vario titolo nello sviluppo del software. Nel 2006 per questioni legali legate alla proprietà del vecchio marchio, ethereal cambia nome e diventa Wireshark.
Questo programma mette a disposizione dell'utente servizi simili ad altri software di analisi pacchetti come tcpdump, ma offrendo un'interfaccia grafica intuitiva e un potente sistema di filtri per ordinare i pacchetti, che lo rendono il più semplice e completo analizzatore di rete disponibile.
Wireshark offre inoltre l'importante opzione di utilizzare la scheda di rete in modalità promiscua permettendo al programma di ascoltare tutto il traffico presente sul punto di ascolto, e non solo quello inviato verso un particolare indirizzo.
La scelta del punto di ascolto è appunto un aspetto molto importante del packet sniffing, specialmente sulle attuali rete switch-ethernet, dove a causa di infrastrutture come switch o router il traffico viene suddiviso e spedito solamente ai nodi che lo richiedono. Esiste quindi il rischio di posizionare il punto di ascolto in una zona non attraversata dalle comunicazioni che si intende ascoltare rendendo cosi inutile la cattura dei pacchetti. In questo caso è necessario modificare il punto di ascolto e/o utilizzare particolari hardware come switch con porta di mirroring, i network tap, o gli ormai introvabili Hub di adeguata velocità per catturare il traffico desiderato.

Il funzionamento di Wireshark per quanto riguarda la cattura dei pacchetti si basa sull'utilizzo delle librerie Pcap (libpcap o il suo porting in windows WinPcap) che offrono un'interfaccia di programmazione dedicata alla cattura del traffico di rete. Quindi la cattura del traffico è possibile solo sulle reti supportate dalla librerie Pcap.
L'analisi e la decodifica dei pacchetti catturati si basa invece su codice proprietario che offre le funzioni di acquisizione e disassemblamento in tempo reale su rete attiva oppure permette di salvare i dati catturati in un file e rielaborarli in un secondo momento.
L'analisi dei pacchetti può essere grafica o da riga di comando, ed è possibile eseguire il riordino dei pacchetti tramite filtri preinstallati o crearne di propri.
Con i "dissector" preinstallati è possibile scomporre e analizzare in maniera mirata le comunicazioni effettuate su ciascuno dei centinaia di protocolli di comunicazione supportati.
Questo fa di wireshark un software particolarmente apprezzato nel suo campo, in particolar modo per la possibilità di riconosce il protocollo utilizzato (almeno fino ad un certo punto...) e tradurre

in forma grafica le informazioni tipiche del protocollo, cosi come può essere l'header dei pacchetti ethernet ip o l'header dei pacchetti SERCOS. In questo modo di semplifica notevolmente la ricerca di informazioni sensibili all'interno della comunicazione.
Nell'ultima versione Wireshark 1.6.8 rilasciata in data 22 giugno 2012, il programma supporta 25 famiglie di protocolli, 33 protocolli proprietari esistenti al di fuori delle famiglie ed esegue catture su 15 livelli fisici differenti.

L'utilizzo di questo software ha portato ad affrontare la problematica relativa al livello di precisione temporale raggiungibile nella cattura dei pacchetti e la conseguente scelta della piattaforma migliore per fare queste misure.
Wireshark infatti non garantisce prestazioni temporali sulla cattura, ma delega la definizione di questa precisione a tre aspetti caratteristici della cattura:

- L'hardware utilizzato per la misura
- Le librerie libpcap/WinPcap installate
- Il sistema operativo in uso, o in maniera più specifica lo scheduler del sistema.

Nel caso in esame l'hardware utilizzato è un Portatile Acer Aspire 5750G con le seguenti caratteristiche:
-Processore Intel Core i7-2630QM operante alla frequenza di 2.0Ghz su 8 core (4 fisici e 4 virtuali)
-4 Gb di memoria DDR3
-Disco fisso 750 Gb 5400rpm
-scheda di rete Broadcom NetLink modello BCM57785

La librerie disponibili al momento della tesi e quindi utilizzate sono: WinPcap 4.1.2 rilasciata il 2 luglio 2010 per quanto riguarda l'ambiente windows e LibPcap 1.3.0 rilasciata il 12 giugno 2012 per sistemi Unix.
I sistemi operativi presi in esame sono stati: Windows 7, Windows Xp, Ubuntu 10.4 standard e con kernel real time, Ubuntu 11.10, Ubuntu 12.10.
Sono stati quindi svolti una serie di test aggiornando sempre i driver hardware all'ultima versione disponibile e mantenendo il sistema scarico durante la cattura. I risultati ottenuti si possono così sintetizzare:

- Le libreria per sistemi Unix sono aggiornate con maggior frequenza, ma non differiscono molto in prestazioni dalle controparti windows.
- La librerie offrono delle prestazioni diverse a seconda del sistema operativo in uso e svolgono al meglio il loro lavoro se supportate da kernel con caratteristiche real time.
- I sistemi basati su Linux anche utilizzando dei kernel modificati espressamente per real-time non hanno comportamenti molto diversi dai software basati su windows.

In pratica, la differenza di prestazioni registrata è sempre stata minima e in ogni caso non decisiva per scegliere una piattaforma piùttosto che un'altra, questo è causato dal fatto che la precisione è limitata prima di tutto dalle prestazioni della scheda di rete che offre una precisione dell'ordine dei 10-100 ms, buona per utilizzi civili ma inadatta a misure industriali dove è richiesta la precisione di 1-10 μs.
Inoltre con l'aumentare delle caratteristiche hardware della macchina, la differenza di prestazioni cambiando schedulatore si assottiglia diventando quasi ininfluente in condizioni di installazione ottimale, cioè: un solo programma installato nel sistema operativo, nessun software di protezione o monitoraggio in esecuzione (in particolare nessun antivirus), disattivazione delle funzioni windows di indicizzazione dei file etc. nessuna applicazione aperta oltre a wireshark.

Nella ricerca di una soluzione al problema delle misure temporali si è cercato un nuovo hardware adatto allo scopo. A tale proposito sono state contattate alcune ditte quali Hilsher, Automata, etc. che offrono schede di rete e interi pacchetti hardware e software specifici per lo studio delle reti. In particolare è stato preso in esame il NetAnalyzer prodotto da hilsher che garantisce un time-stamp dell'ordine del ηs, ma il prezzo richiesto di circa 1700 euro ha fatto desistere dall'acquisto.

Capitolo 6

Analisi sperimentale delle comunicazioni

In questo capitolo verrà esposta la prima parte del lavoro eseguito, ovvero l'analisi dei pacchetti effettivamente circolanti nelle due reti.

Questa analisi viene condotta utilizzando il software proprietario dell'impianto al semplice scopo di accendere la rete e generare del traffico. Parallelamente, attraverso il software Wireshark, si cattura questo traffico e si ricostruisce la logica della comunicazione.

6.1 SERCOS III

L'impianto Bosch Rexroth al momento dell'accensione attraversa, come già visto nella teoria 4 fasi; l'analisi di questo start-up della rete fornisce informazioni molto importanti per lo studio successivo dei pacchetti. Si ripercorrono quindi queste fasi descrivendo i pacchetti circolanti in rete e le loro funzioni.

Durante la prima fase (CP0) la struttura della comunicazione catturata è formata da due pacchetti: Il master spedisce in rete ad ogni ciclo il seguente telegramma:

MDT
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 20 40
c5 e3 1f 81 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00

Come si nota è un telegramma leggero in quanto è formato solo da 60 Byte, e "vuoto" nel senso che non contiene dati. Lo scopo di questo pacchetto è solamente verificare l'integrità della rete e dei dispositivi connessi.

Il passaggio tra fase 0 e fase 1 avviene trasmettendo in rete messaggi con la medesima struttura della fase 0, ma modificando i byte che compongono il campo fase, segnalando in questo modo che è in atto una transazione verso una nuova fase. Questo è un comportamento standard della comunicazione, e si ripeterà ad ogni cambiamento di fase.

Il secondo pacchetto, che compone questo ciclo viene segnalato come errato (Malformed packet) e ricalca la struttura del primo, tranne che per la dimensione, in questo caso 532 byte.

AT
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 60 20
98 cd d4 3c 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...

Mano a mano che questi pacchetti si susseguono, nell secondo pacchetto vengono commutati a

1 due byte nel seguente modo

AT
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 60 70
6c 9c bf 57 00 00 01 00 01 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...

In questo modo gli slave hanno modificato il messaggio per segnalare al master la loro presenza in rete.

In fase 1 la struttura della comunicazione cambia, si alternano in rete 2 MDT della lunghezza di 1300 byte e 2 AT della medesima lunghezza.
Nell'impianto in analisi formato da soli due slave, solamente il primo MDT e AT contengono informazioni, il secondo viene spedito per protocollo, non per reale necessità.
Questi due messaggi sono ancora "vuoti" di dati, tranne per il meccanismo di handshake tra master (byte 01) e slave (byte 09).

MDT0
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 20 11
a7 82 73 9d 00 00 00 00 00 00 01 00 00 00 00 00
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...

AT0
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 60 01
c6 dd bd 70 00 00 00 00 00 00 09 00 00 00 00 00
09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...

La fase 2 è la più importante nella parametrizzazione dell'impianto, in questa fase infatti il master elabora le impostazioni ricevute del software Indraworks, le trasforma in identificativi (IDNs) e le spedisce in rete a tutti i driver presenti. Catturando e interpretando il traffico in questa fase è possibile "leggere" il tempo di ciclo, la struttura del MDT e AT che sarà utilizzata in fase 4, etc.
La struttura dei pacchetti è simile alla fase 1, 2 MDT della lunghezza di 1300 byte e 2 AT della medesima lunghezza.
Analizziamo a titolo di esempio la trasmissione di un identificativo :

MDT0
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 20 72
45 b2 c8 49 00 00 00 00 00 00 0e 00 86 01 00 00
0e 00 86 01 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...

AT0
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 60 72
40 fd b1 b9 00 00 00 00 00 00 0a 00 02 00 0a 00
08 00 00 00 0a 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00...

In questo caso il master comunica sul canale di servizio il controllo ”00 0e” che va interpretato nel modo seguente:
....00 1... = Data block element: Element 1: Opening IDN (1)
.... .1.. = End of element transmission: Last transmission (1)
....1. = Read/Write: Write SVC INFO (1)
....0 = Master Handshake: 0

Il campo info del canale di servizio contiene il valore ”86 01 00 00” che invertito (00 00 01 86) e convertito in decimale (390) indica che i pacchetti che seguiranno conterranno i data bloc element riferiti all'identificativo s-0-0390.
A seconda del tipo di identificativo invocato, possono essere trasferiti agli slave, sempre utilizzando il canale di servizio; delle stringhe, valori interi, liste di IDNs, strutture, etc... Nell'impianto in analisi questa parte della parametrizzazione è di difficile lettura in quanto ogni comando viene spedito più volte, e gli slave spesso rifiutano l'handshake del pacchetto perchè impegnati in altre procedure. Questo ha obbligato a leggere diverse centinaia di pacchetti solo per concludere la trasmissione di un IDNs.

Terminata la fase 2, con una transizione si passa alla fase 3 e inizia la comunicazione real time.
I pacchetti si snelliscono e la comunicazione si riduce a un solo MDT e un solo AT entrambi della lunghezza di 96 byte. La struttura di questi pacchetti ricalca quella analizzata nello studio teorico della rete, sono quindi presenti i campi hot-plug, device status, device control, etc. ma non essendo ancora entrato in servizio l'impianto, il campo data è ”vuoto”. Continua invece la parametrizzazione della rete attraverso il canale di servizio.

MDT0:
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 20 53
1b a2 a1 05 00 00 00 00 00 00 00 00 0f 00 80 00
00 00 0e 00 80 00 00 00 00 00 00 00 00 01 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

AT0:
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 60 53
1e ed d8 f5 00 00 00 00 00 00 00 00 09 00 07 00
03 00 08 00 07 00 03 00 10 08 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 10 08 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Con un'ultima transizione si accede alla fase 4, dove tutti gli identificativi sono stati correttamente ricevuti dagli slave ed è possibile utilizzare l'impianto. Questa fase rappresenta il funzionamento a regime della comunicazione SERCOS III, le precedenti sono servite solamente a ”preparare” i dispositivi dell'impianto.

Si entra ora nel dettagli dell'impianto in analisi; nei test effettuati sono stati creati programmi per il controllo in posizione dei motori, il gearing e la camma virtuale. Questi programmi hanno generato a loro volta nell'impianto due strutture diverse per i pacchetti, di dimensioni diverse e formati da identificativi diversi.

Nello specifico i programmi che contenevano strutture di movimentazione assi come Move Absolute o il gearing generavano una pacchetto di questo tipo:

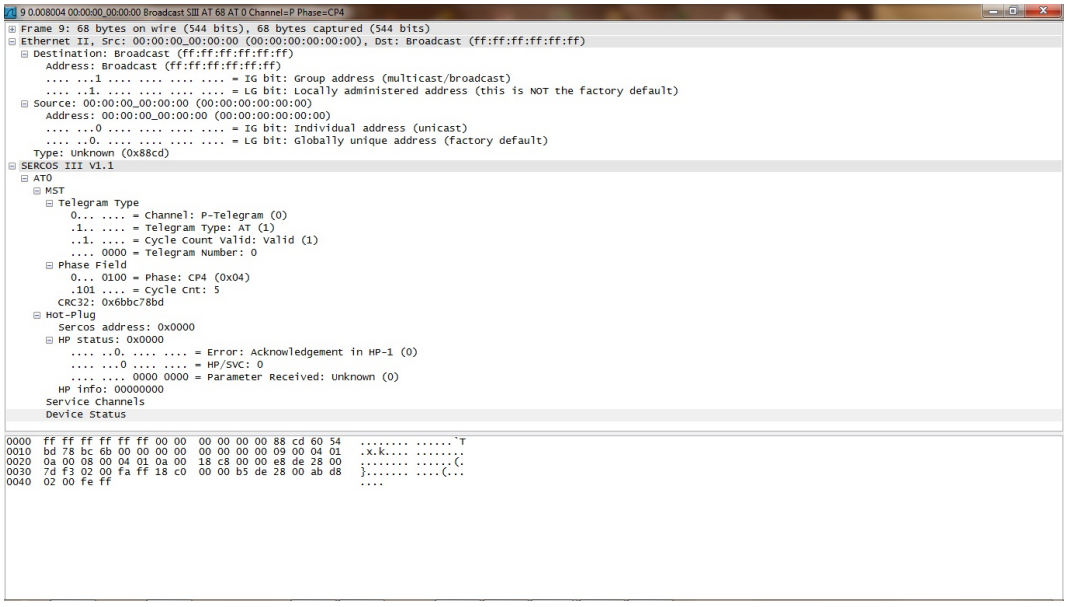


Fig. 6.1: Esempio pacchetto SERCOS III

I byte che compongono la comunicazione sono:
AT0 (68 byte)
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 60 54
bd 78 bc 6b 00 00 00 00 00 00 00 09 00 04 01
0a 00 08 00 04 01 0a 00 18 c8 00 00 e8 de 28 00
7d f3 02 00 fa ff 18 c0 00 00 b5 de 28 00 ab d8
02 00 fe ff

Il loro significato è:

- ff ff ff ff ff ff ⇒ indirizzo di destinazione (broadcast).
- 00 00 00 00 00 00 ⇒ indirizzo sorgente non utilizzato.
- 88 cd ⇒ EtherType SERCOS III
- 60 ⇒ 0110 0000
  0 = P-telegram
  1 = telegramma AT
  1 = Conteggio ciclo valido
  0000 = telegramma numero 0

- 54 ⇒ 0101 0100
  0... 0100 = fase 4
  .101 .... = conteggio ciclo: 5

- bd 78 bc 6b ⇒ codice ridondanza ciclica.

- 00 00 00 00 00 00 00 00 ⇒ hot-plugin: nessun nuovo dispositivo desidera entrare in rete

- 09 00 04 01 0a 00: service channel 1.
  09 00
  .... 1... = SVC process: SVC valid (1)
  .... .0.. = SVC Error: No error (0)
  ....0. = Busy: Step finished, slave ready for new step (0)
  ....1 = Handshake: 1
  I restanti campi non sono utilizzati in questo tipo di comunicazione

- 08 00 04 01 0a 00 = service channel 2. 08 00
  .... 1... = SVC process: SVC valid (1)
  .... .0.. = SVC Error: No error (0)
  ....0. = Busy: Step finished, slave ready for new step (0)
  ....0 = Handshake: 0
  I restanti campi non sono utilizzati in questo tipo di comunicazione

- 18 c8 00 00 ⇒ c8 18 ⇒ 1100100000011000 status word slave 1.
  11 = (bit 14 15) In funzionamento, l'azionamento è in coppia
  0 = (bit 13) Driver non bloccato.
  0 = (bit 12) cambio bit diagnostica classe 2.
  1 = (bit 11) cambio bit diagnostica classe 3.
  00 = (bit 8-9) operazione in modalità primaria.
  00 = (bit 7-6) bit di stato Real-Time.
  0 = (bit 5) comando cambio bit.
  000 = (bit 0-1-2) informazioni su canale di servizio.

- e8 de 28 00 7d f3 02 00 fa ff ⇒ dati slave 1.
  e8 de 28 00⇒ Posizione dispositivo 1 (S-0-0051)
  7d f3 02 00⇒ velocità dispositivo 1 (S-0-0040)
  fa ff ⇒ coppia dispositivo 1 (S-0-0084)

- 18 c0 00 00 ⇒ c0 18 ⇒ 1100000000011000 status word slave 2. 11 = (bit 14 15) In funzionamento, l'azionamento è in coppia
  0 = (bit 13) Driver non bloccato.
  0 = (bit 12) cambio bit diagnostica classe 2.
  0 = (bit 11) cambio bit diagnostica classe 3.
  00 = (bit 8-9) operazione in modalità primaria.
  00 = (bit 7-6) bit di stato Real-Time.
  0 = (bit 5) comando cambio bit.
  000 = (bit 0-1-2) informazioni su canale di servizio.

- b5 de 28 00 ab d8 02 00 fe ff ⇒ dati slave 2.
  b5 de 28 00 ⇒ Posizione dispositivo 1 (S-0-0051)
  ab d8 02 00 ⇒ Velocità dispositivo 1 (S-0-0040)
  fe ff ⇒ Coppia dispositivo 1 (S-0-0084)

MDT0 (60 Byte)
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 20 24
84 46 c0 cb 00 00 00 00 00 00 00 00 3d 00 86 01
00 00 3c 00 86 01 00 00 00 e0 00 00 1f d3 28 00
00 e0 00 00 1f d3 28 00 00 00 00 00

- ff ff ff ff ff ff ⇒ indirizzo di destinazione (broadcast).
- 00 00 00 00 00 00 ⇒ indirizzo sorgente non utilizzato.
- 88 cd ⇒ EtherType SERCOS III
- 20 ⇒ 0010 0000
  0... .... = Channel: P-Telegram (0)
  .0.. .... = Telegram Type: MDT (0)
  ..1. .... = Cycle Count Valid: Valid (1)
  .... 0000 = Telegram Number: 0
- 24 ⇒ 0010 0100
  0... 0100 = Phase: CP4 (0x04)
  .010 .... = Cycle Cnt: 2
- 84 46 c0 cb ⇒ codice ridondanza ciclica.
- 00 00 00 00 00 00 00 00 ⇒ hot-plugin: nessun nuovo dispositivo desidera entrare in rete
- 3d 00 86 01 00 00 ⇒ service channel 1.
- 3c 00 86 01 00 00 ⇒ service channel 2.
- 00 e0 00 00 ⇒ control word device 1.
- 1f d3 28 00 ⇒ Posizione di riferimento dispositivo 1 (S-0-0047)
- 00 e0 00 00 ⇒ control word device 2.
- 1f d3 28 00 ⇒ Posizione di riferimento dispositivo 2 (S-0-0047)
- 00 00 00 00 00 ⇒ non utilizzati.

Mentre inserendo la funzione camma elettronica, il messaggio diventa più lungo e più complicato. Cambiano gli identificativi che formano il pacchetto e viene utilizzata una struttura chiamata data-container per trasportare i dati. Questo significa che ad ogni posizione nel messaggio non viene più associato una particolare funzione come posizione,velocità etc, bensì una lista di dati possibili e a seconda dell'indice che viene scelto si hanno composizioni diverse del pacchetto.
Per capire quali dati sono effettivamente contenuti nel messaggio è necessario capire prima il meccanismo utilizzato per creare, trasferire e indicizzare le liste.
In questo campo è stata provvidenziale la lettura del documento tecnico "Rexroth IndraDrive Drive Controllers MPx-02; MPx-03; MPx-04" che elenca ordinatamente tutti gli identificativi della

rete SERCOS. É quindi stato possibile attraverso la cattura degli identificativi nella parametrizzazione, la loro traduzione (utilizzando questo documento) comprendere il numero di liste utilizzato, l'indice in uso e infine, la struttura "nascosta" del pacchetto.

I data container sono dei puntatori a delle liste di IDNs, cioè puntano a un'altro identificativo che contiene varie funzioni, quali feedback velocità , comando posizione, limite coppia, etc.
L'IDNs S-0-0368, presente in tutti i messaggi del tipo data container, è l'indice di queste liste, più precisamente i primi 4 byte di questo campo è l'indice per i pacchetto MDT e gli ultimi 4 l'indice per il pacchetto AT.
Per rendere più chiaro il procedimento seguito si analizza un pacchetto.

MDT: (dimensioni 96 byte)
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 20 34
e0 56 77 d6 00 00 00 00 00 00 00 00 3c 00 a8 01
00 00 3d 00 43 86 00 00 00 e1 00 00 05 01 09 0e
00 00 00 00 40 0d 03 00 10 27 00 00 20 4e 00 00
50 c3 00 00 00 e1 00 00 05 01 01 00 00 00 00 00
40 0d 03 00 10 27 00 00 20 4e 00 00 50 c3 00 00

- ff ff ff ff ff ff ⇒ indirizzo di destinazione (broadcast).
- 00 00 00 00 00 00 ⇒ indirizzo sorgente non utilizzato.
- 88 cd ⇒ EtherType SERCOS III
- 20 ⇒ 0010 0000
  0... .... = Channel: P-Telegram (0)
  .0.. .... = Telegram Type: MDT (0)
  ..1. .... = Cycle Count Valid: Valid (1)
  .... 0000 = Telegram Number: 0
- 34 ⇒ 0010 0100
  0... 0100 = Phase: CP4 (0x04)
  .100 .... = Cycle Cnt: 3
- e0 56 77 d6 ⇒ codice ridondanza ciclica.
- 00 00 00 00 00 00 00 00 ⇒ hot-plugin: nessun nuovo dispositivo desidera entrare in rete
- 3c 00 a8 01 00 00 ⇒ service channel 1.
- 3d 00 43 86 00 00 ⇒ service channel 2.
- 00 e1 00 00 ⇒ Control word device1.
- 05 01 ⇒ questo campo indica gli indici delle liste (S-0-0368).L'indice del MDT in questo caso è 5, mentre l'indice AT è 1
- 09 0e ⇒ Signal Control Word (S-0-0145). Si tratta di 2 byte utilizzati per la comunicazione real time tra controllore e driver.

- 00 00 00 00 ⇒ Master Axis Position (P-0-0053). Il Master comunica la posizione dell'asse virtuale per permettere la sincronizzazione degli slave.

- 40 0d 03 00 ⇒ Data container A: command value 1 (S-0-0360). Primo contenitore di dati, è necessario accedere alla lista S-0-0370 che al 5 elemento contiene (S-0-0282) Positioning command value.

- 10 27 00 00 ⇒ Data container A: command value 2 (S-0-0450), si accede alla lista S-0-0490 e si preleva l'elemento S-0-0259 Positioning deceleration.

- 20 4e 00 00 ⇒ Data container A: command value 3 S-0-0451 legato alla lista s-0-0491 che contiene S-0-0260 Positioning acceleration.

- 50 c3 00 00 ⇒ Data container A: command value 4 S-0-0452 legato alla lista s-0-0492 che contiene S-0-0359 Positioning deceleration.

- La medesima analisi vale per il drive 2.
  00 e1 00 00 ⇒ Control word device2.
  05 01 ⇒ Indici liste (S-0-0368)
  01 00 ⇒ Signal Control Word (S-0-0145)
  00 00 00 00 ⇒ Master Axis Position (P-0-0053)
  40 0d 03 00 ⇒ Positioning command value (S-0-0282).
  10 27 00 00 ⇒ Positioning deceleration (S-0-0259).
  20 4e 00 00 ⇒ Positioning acceleration (S-0-0260).
  50 c3 00 00 ⇒ Positioning deceleration (S-0-0359).

AT: (dimensioni 96 byte)
ff ff ff ff ff ff 00 00 00 00 00 00 88 cd 60 34
e5 19 0e 26 00 00 00 00 00 00 00 00 08 00 00 00
00 00 09 00 00 00 00 00 18 c9 00 00 05 01 09 0e
00 00 00 00 7c fb 9e 0d 7c 3f 03 00 00 00 00 00
01 01 0a 00 18 c9 00 00 05 01 09 0e 00 00 00 00
2f a4 c8 11 f5 1d 03 00 14 00 00 00 01 01 0a 00

- ff ff ff ff ff ff ⇒ indirizzo di destinazione (broadcast).

- 00 00 00 00 00 00 ⇒ indirizzo sorgente non utilizzato.

- 88 cd ⇒ EtherType SERCOS III

- 60 ⇒ 0110 0000
  0... .... = Channel: P-Telegram (0)
  .1.. .... = Telegram Type: AT (1)
  ..1. .... = Cycle Count Valid: Valid (1)
  .... 0000 = Telegram Number: 0

- 34 ⇒ 0010 0100
  0... 0100 = Phase: CP4 (0x04)
  .100 .... = Cycle Cnt: 3

- e5 19 0e 26 ⇒ codice ridondanza ciclica.

- 00 00 00 00 00 00 00 00 ⇒ hot-plugin: nessun nuovo dispositivo desidera entrare in rete
- 08 00 00 00 00 00 ⇒ service channel 1.

- 09 00 00 00 00 00 ⇒ service channel 2.

- 18 c9 00 00 ⇒ Status word device 1.
- 05 01 ⇒ Indici liste (S-0-0368) in questo caso si utilizza 1
- 09 0e ⇒ Signal Status Word (S-0-0144). Si tratta di 2 byte utilizzati per la comunicazione real time tra controllore e driver.
- 00 00 00 00 ⇒ Actual position value of measuring encoder (P-0-0052).
- 7c fb 9e 0d ⇒ Data container A: feedback value 1 (S-0-0364) legato alla lista S-0-0371 contiene al primo posto Active position feedback value (S-0-0386).
- 7c 3f 03 00 ⇒ Data container A: feedback value 2 (S-0-0480) legato alla lista S-0-0500 contiene al primo posto Velocity feedback value (S-0-0040).
- 00 00 00 00 ⇒ Data container A: feedback value 3 (S-0-0481) legato alla lista S-0-0501 contiene al primo posto Torque/force feedback value (S-0-0084).
- 01 01 0a 00 ⇒ Data container A: feedback value 4 (S-0-0482) non contiene dati.
- La medesima analisi vale per il drive 2.
  18 c9 00 00 ⇒ Status word device 2.
  05 01 ⇒ Indici liste
  09 0e ⇒ Signal Status Word (S-0-0144)
  00 00 00 00 ⇒ Actual position
  2f a4 c8 11 ⇒ Active position feedback (S-0-0386)
  f5 1d 03 00 ⇒ Velocity feedback (S-0-0040)
  14 00 00 00 ⇒ Torque/force feedback (S-0-0084)
  01 01 0a 00 ⇒ feedback value 4 (S-0-0482)

La Strutture di tutti i pacchetti sono state comprese a fondo, questo permetterà nella seconda parte della tesi di accedere ai dati in modo ordinato ed eseguire l'analisi della sincronizzazione.

6.2 EtherNet/IP

L'analisi dei pacchetti EtherNet/IP ha dovuto essere forzatamente più breve, l'impianto è stato assemblato un mese e mezzo prima dei termini del lavoro, ed è stato possibile fare solo un'analisi ridotta del protocollo.
Nello specifico si è concentrata l'attenzione solamente sui telegrammi che circolano in rete successivamente alla parametrizzazione dell'impianto, e si sono analizzati i messaggi spediti dagli slave al master.
Si inizia ora l'analisi dal pacchetto più semplice, generato dall'impianto durante una procedura di

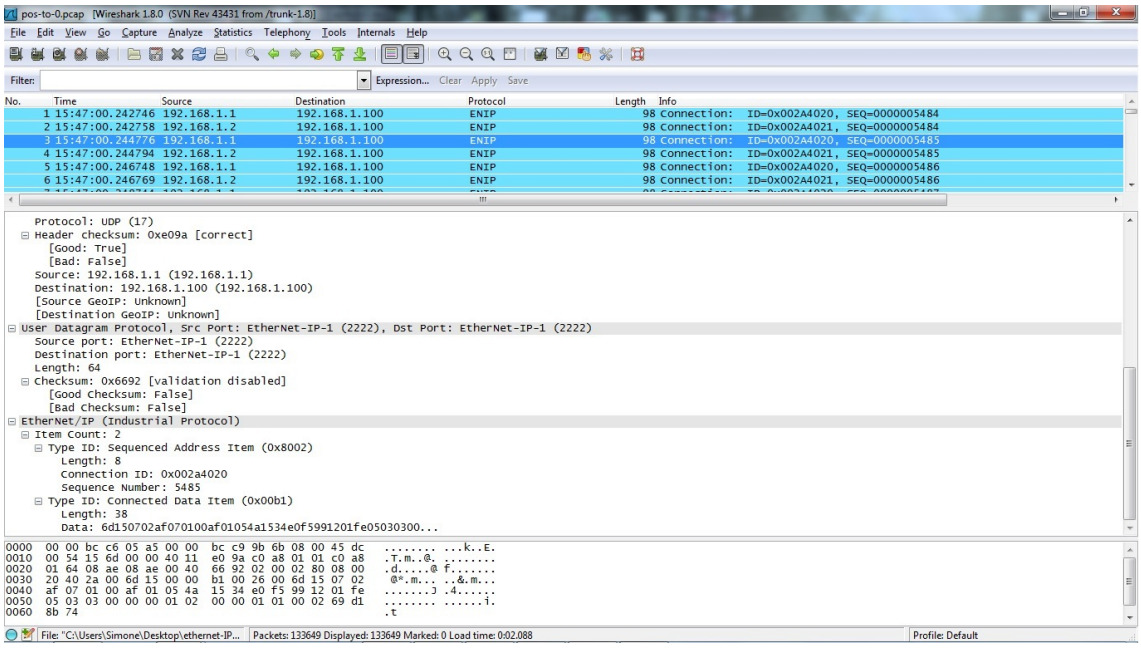


Fig. 6.2: Cattura EtherNet/IP

controllo in posizione di entrambi gli assi, senza sincronizzazione degli assi attiva.

Pacchetto Slave drive 1 (98 byte)

00 00 bc c6 05 a5 00 00 bc c9 9b 6b 08 00 45 dc
00 54 53 a8 00 00 40 11 a2 5f c0 a8 01 01 c0 a8
01 64 08 ae 08 ae 00 40 66 5e 02 00 02 80 08 00
d9 40 13 00 a8 53 00 00 b1 00 26 00 a8 53 07 02
e6 07 01 00 e6 01 02 58 2c 90 60 22 90 12 01 0e
05 03 03 00 00 00 01 02 00 00 01 01 00 04 e1 53
04 00

Ethernet header:

- 00 00 bc c6 05 a5: Il MAC address del destinatario del messaggio.
- 00 00 bc c9 9b 6b: Il MAC address del mittente.
- 08 00 Viene utilizzato il protocollo Internet Protocol, Version 4 (IPv4)

Internet Protocol Header Data:

- 4: La versione IP utilizzata è IP 4
- 5: La lunghezza dell'header è di 20 byte.
- dc: Indica un livello di servizio (0x37)
  1101 11.. = Differentiated Services Codepoint: Unknown (0x37
  .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport)
- 00 54: La lunghezza totale del pacchetto è di 84 Byte.
- 53 a8: L'identificatore ID del messaggio è 21416.

- 00 00: Il messaggio non è stato frammentato.
- 40: Il pacchetto può attraversare ancora 64 dispositivi.
- 11: Il protocollo della parte successiva è UDP (17).
- a2 5f Campo di sicurezza
- c0 a8 01 01: L'indirizzo IP del mittente del messaggio è 192.168.1.1
- c0 a8 01 64: L'indirizzo IP del destinatario del messaggio è 192.168.1.100

UDP:

- 08 ae: La porta del dispositivo mittente è EtherNet-IP-1 (2222).
- 08 ae: La porta del dispositivo destinatario è EtherNet-IP-1 (2222).
- 00 40: La lunghezza totale del messaggio è 64 byte.
- 66 5e: Campo di sicurezza

Header dal Protocollo CIP:

- 02 00: Come sempre nel protocollo UDP CIP questo valore è 2.
- 02 80: tipo Sequenced addressed item.
- 08 00: Lunghezza 8 byte
- d9 40 13 00: Identificatore di connessione 00 13 40 d9
- a8 53 00 00: Numero sequenziale: 21416
- b1 00: Tipo di dato connesso.
- 26 00: Indica la lunghezza del campo dati. in questo caso 38 byte.
- a8 53 07 02 e6 07 01 00 e6 01 02 58 2c 90 60 22 90 12 01 0e 05 03 03 00 00 00 01 02 00 00 01 01 00 04 e1 53 04 00 campo dati.

Campo dati CIP:

- a8 53: (contatore A) i primi 2 byte sono il contatore dei cicli di comunicazione. Nell'impianto in analisi ad ogni ciclo di comunicazione vengono spediti 4 pacchetti: due dai rispettivi slave verso il master e due dal master verso ciascuno slave. Questo contatore rimane immutato per tutti questi pacchetti, poi al ciclo successivo incrementa di 1.
- 07 02: questi 2 byte non cambiano per tutta la comunicazione.
- e6 :(contatore B) questo campo è un altro contatore che incrementa con la stessa frequenza di A, ma presenta una base del conteggio differente.
- 07 01: Questi 2 byte non cambiano per tutta la comunicazione.
- 00: Altro campo non modificato durante la comunicazione.
- e6: (contatore C) contatore sincronizzato con B.
- 01 02: Questi 2 byte non cambiano per tutta la comunicazione.

- 58 2c 90 : L'informazione contenuta in questo campo non è stata identificata.
- 60: contatore incrementa ogni 10 messaggi
- 22 90 : Questi 2 byte non cambiano per tutta la comunicazione.
- 12 01 0e 53 03 03 00 00 00 01 02 00 00 01 01 00 02: questa sequenza di byte è riferita agli I/O aggiuntivi presenti nel driver, non essendo però utilizzati, questo campo rimane invariato per tutta la comunicazione.
- e1 53 04 00: Posizione corrente del drive espressa in impulsi totali dell'encoder.

Il pacchetto spedito al master dallo slave 2 ha la medesima struttura, vengono ovviamente sostituiti gli indirizzi del mittente e i codici di sicurezza CRC.

Questa tipologia di pacchetto appena analizzato è la più breve e semplice tra quelle incontrate nelle varie catture. Infatti a seconda di quali funzioni vengono richiamate dal programma, anche in questo impianto assistiamo alla modifica della struttura del pacchetto.
La seconda tipologia di pacchetti catturata in rete compare quando si inserisce nel programma RSLogix 5000 una delle seguenti funzioni:
-funzione di gearing tra l'asse master virtuale e i due assi reali.
-funzione di gearing tra un asse reale master e un asse reale slave.
-funzione di camma elettronica sui due slave.

In questo caso i pacchetti cambiano forma e dimensione; lo slave 1 produce un messaggio di dimensioni 126 byte, e lo slave 2 un pacchetto di 114 byte.

Pacchetto slave drive2 (114 byte)
00 00 bc c6 05 a5 00 00 bc c9 9b 34 08 00 45 dc
00 64 71 f5 00 00 40 11 84 01 c0 a8 01 02 c0 a8
01 64 08 ae 08 ae 00 50 e8 61 02 00 02 80 08 00
14 40 39 00 f5 71 00 00 b1 00 36 00 f5 71 07 02
2d 07 01 00 2d 09 76 57 17 35 16 00 9d 12 12 e9
11 35 16 00 9d 12 6a 07 1e 35 16 00 9d 12 01 48
05 03 03 00 00 00 01 02 00 00 01 01 00 04 31 b2
63 fe

La prima parte, l' Ethernet header rimane invariato

- 00 00 bc c6 05 a5: Il MAC address del destinatario del messaggio.
- 00 00 bc c9 34 08: Il MAC address del mittente, cambia rispetto allo scorso pacchetto perchè il mittente è il drive 2
- 08 00: Viene utilizzato il protocollo Internet Protocol, Version 4 (IPv4)

Campo dati header:

- 4: La versione IP utilizzata è IP 4
- 5: La lunghezza dell'header è di 20 byte.
- dc: Indica un livello di servizio (0x37)
  1101 11.. = Differentiated Services Codepoint: Unknown (0x37) .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport)

- 00 64: La lunghezza totale del pacchetto è di 100 Byte.
- 71 f5: L'identificatore ID del messaggio è 29173.
- 00 00: Il messaggio non è stato frammentato.
- 40: Il pacchetto può attraversare ancora 64 dispositivi.
- 11: Il protocollo della parte successiva è UDP (17).
- 84 01: Campo di sicurezza
- c0 a8 01 02: L'indirizzo IP del mittente del messaggio è 192.168.1.2
- c0 a8 01 64: L'indirizzo IP del destinatario del messaggio è 192.168.1.100

UDP:

- 08 ae: La porta del dispositivo mittente è EtherNet-IP-1 (2222).
- 08 ae: La porta del dispositivo destinatario è EtherNet-IP-1 (2222).
- 00 50: La lunghezza totale del messaggio è 80 byte.
- e8 61: Campo di sicurezza

Header dal Protocollo CIP:

- 02 00: Come sempre nel protocollo UDP CIP questo valore è 2.
- 02 80: tipo Sequenced addressed item.
- 08 00: Lunghezza 8 byte
- 14 40 39 00: Identificatore di connessione 00 14 40 39
- f5 71 00 00: Numero sequenziale: 29173
- b1 00: Tipo di dato connesso.
- 36 00: Indica la lunghezza del campo dati. in questo caso 54 byte.
- f5 71 07 02 2d 07 01 00 2d 09 76 57 17 35 16 00 9d 12 12 e9 11 35 16 00 9d 12 6a 07 1e 35 16 00 9d 12 01 48 05 03 03 00 00 00 01 02 00 00 01 01 00 04 31 b2 63 fe: campo dati.

Campo dati CIP:

- f5 71: (contatore A) contatore di cicli.
- 07 02: invariato per tutta la trasmissione.
- 2d: (contatore B) sincronizzato con contatore C.
- 07 01: invariato per tutta la trasmissione.
- 00: invariato per tutta la trasmissione.
- 2d: (contatore C) sincronizzato con B.
- 09: invariato per tutta la trasmissione.
- 76 57 17: informazione incognita

- 35: (contatore D) incrementa ogni 10 messaggi
- 16 00 9d 12: invariato per tutta la trasmissione.
- 12 e9 11: informazione incognita
- 35 (contatore E) sincronizzato con D incrementa ogni 10 messaggi.
- 16 00 9d 12: invariato per tutta la trasmissione.
- 6a 07 1e: informazione incognita
- 35: (Contatore F) sincronizzato con D incrementa ogni 10 messaggi.
- 16 00 9d 12: invariato per tutta la trasmissione.
- 01 48 05 03 03 00 00 00 01 02 00 00 01 01 00 04: questa sequenza di byte è riferita agli I/O aggiuntivi presenti nel driver, non essendo però utilizzati, questo campo rimane invariato per tutta la comunicazione.
- 31 b2 63 fe: Informazione di posizione in impulsi encoder assoluto

Pacchetto slave driver 1 (126 byte)
00 00 bc c6 05 a5 00 00 bc c9 9b 6b 08 00 45 dc
00 70 f8 7c 00 00 40 11 fd 6e c0 a8 01 01 c0 a8
01 64 08 ae 08 ae 00 5c 7e 80 02 00 02 80 08 00
5e 44 40 00 7c f8 00 00 b1 00 42 00 7c f8 07 02
b8 07 01 00 b8 09 85 6b 45 f7 1f 13 00 00 d9 f4
3f f7 1f 13 00 00 7d e9 4c f7 1f 13 00 00 01 cd
08 06 03 03 00 00 01 02 00 00 01 01 00 04 86 d6
12 00 00 00 01 00 c6 e6 12 00 3a d6 12 00

Ethernet header:

- 00 00 bc c6 05 a5: Il MAC address del destinatario del messaggio.
- 00 00 bc c9 9b 6b: Il MAC address del mittente.
- 08 00: Viene utilizzato il protocollo Internet Protocol, Version 4 (IPv4)

Internet Protocol Header Data:

- 4: La versione IP utilizzata è IP 4
- 5: La lunghezza dell'header è di 20 byte.
- dc: Indica un livello di servizio (0x37)
  1101 11.. = Differentiated Services Codepoint: Unknown (0x37)
  .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport)
- 00 70: La lunghezza totale del pacchetto è di 84 Byte.
- f8 7c: L'identificatore ID del messaggio è 63612.
- 00 00: Il messaggio non è stato frammentato.
- 40: Il pacchetto può attraversare ancora 64 dispositivi.

- 11: Il protocollo della parte successiva è UDP (17).
- fd 6e: Campo di sicurezza
- c0 a8 01 01: L'indirizzo IP del mittente del messaggio è 192.168.1.1
- c0 a8 01 64: L'indirizzo IP del destinatario del messaggio è 192.168.1.100

UDP:

- 08 ae: La porta del dispositivo mittente è EtherNet-IP-1 (2222).
- 08 ae :La porta del dispositivo destinatario è EtherNet-IP-1 (2222).
- 00 5c: La lunghezza totale del messaggio è 92 byte.
- 7e 80: Campo di sicurezza

Header dal Protocollo CIP:

- 02 00: Come sempre nel protocollo UDP CIP questo valore è 2.
- 02 80: tipo Sequenced addressed item.
- 08 00: Lunghezza 8 byte
- 5e 44 40 00: Identificatore di connessione 00 40 44 5e
- 7c f8 00 00: Numero sequenziale: 63612
- b1 00: Tipo di dato connesso.
- 42 00: Indica la lunghezza del campo dati. in questo caso 66 byte.
- 7c f8 07 02 b8 07 01 00 b8 09 85 6b 45 f7 1f 13 00 00 d9 f4 3f f7 1f 13 00 00 7d e9 4c f7 1f 13 00 00 01 cd 08 06 03 03 00 00 01 02 00 00 01 01 00 04 86 d6 12 00 00 00 01 00 c6 e6 12 00 3a d6 12 00: campo dati.

Campo dati CIP:

- 7c f8: (contatore A) ciclo di comunicazione
- 07 02: questi 2 byte invariati per tutta la comunicazione.
- b8 :(contatore B) contatore sincrono con A, ma con offset presente
- 07 01: Questi 2 byte non cambiano per tutta la comunicazione.
- 00: Altro campo non modificato durante la comunicazione.
- b8: (contatore C) contatore sincronizzato con B
- 09 85: Questi 2 byte non cambiano per tutta la comunicazione.
- 6b 45 7f 1f : L'informazione contenuta in questo campo non è stata identificata
- 13: (contatore D) questo contatore incrementa ogni 10 messaggi
- 00 00: invariato per tutta la trasmissione
- d9 f4 3f f7: informazione incognita

- 13: (contatore E) incrementa ogni 10 messaggi sincronizzato con D,F
- 00 00: invariato per tutta la trasmissione
- 7d e9 4c f7 1f : Informazione incognita
- 13: (contatore F) incrementa ogni 10 messaggi sincronizzato con D,E
- 00 00: invariato per tutta la trasmissione
- 01 cd 08 06 03 03 00 00 01 02 00 00 01 01 00 04: questa sequenza di byte è riferita agli I/O aggiuntivi presenti nel driver, non essendo però utilizzati, questo campo rimane invariato per tutta la comunicazione.
- 86 d6 12 00: posizione utilizzata dal controllore per pianificare la sincronizzazione
- 00 00 01 00: invariato per tutta la trasmissione
- c6 e6 12 00: posizione utilizzata dal controllore per pianificare la sincronizzazione
- 3a d6 12 00: posizione corrente asse in impulsi encoder assoluto

Capitolo 7

Analisi degli impianti

Per lo studio delle reti SERCOS III e EtherNet/IP sono stati utilizzati due banchi prova presenti in università e configurati entrambi nel seguente modo:

-Un controllore che opera da master della rete.

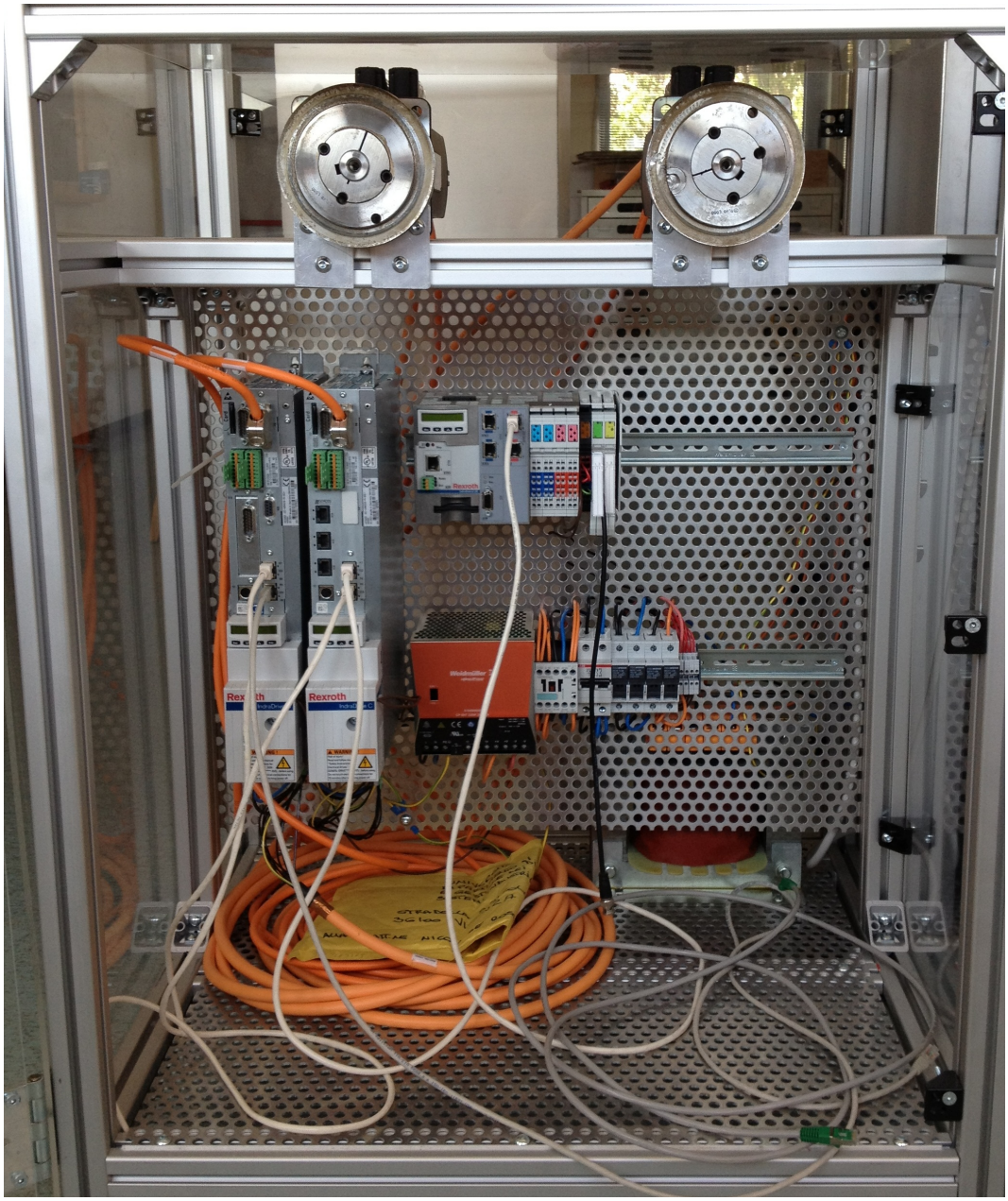
-Due drive connessi in topologia lineare tramite un cavo cat5e della lunghezza approssimativa di 3 metri ciascuno.

-Due motori comandati direttamente dal rispettivo driver.

A questa dotazione si affianca il portatile descritto nel capitolo 5 e, nel solo caso della rete EtherNet/IP, un HUB a 100 Mbps utilizzato per creare un punto di ascolto dove catturare i pacchetti circolanti in rete.

É stata scelta questa configurazione comune per i due impianti in quanto lascia l'ultimo slave della rete con una porta inutilizzata, permettendo quindi di eseguire le catture.
In secondo luogo topologie diverso come l'anello in SERCOS modificano la struttura della comunicazione aumentando il numero di pacchetti circolanti in rete o strutture come la stella in EtherNet/IP richiedono hardware aggiuntivo non disponibile.
Infine mantenendo la medesima struttura nei due impianti è possibile analizzare i risultati dei test di motion control in maniera quanto più omogenea possibile tra i due impianti.
Questa omogeneità si è però dovuta limitare alla configurazione topologica della rete, in quanto gli azionamenti e i motori Bosch Rexroth hanno caratteristiche diverse rispetto a quelli Allen Bradley, e non sono interscambiabili.
Questi limiti pratici ha portato a intensificare lo studio degli aspetti di comunicazione tra i dispositivi e la parametrizzazione della rete attraverso il software proprietario piuttosto che a misure prettamente meccaniche riguardanti la performance dei motori.

7.1 L'impianto Bosch Rexroth



L'impianto Bosch Rexroth utilizzato durate le prove è formato da un controllore Rexroth IndraControl L65, 2 azionamenti IndraDrive C - compact converters, e 2 motori sincroni tipo Msk 060c.

Il controllore L65 rappresenta una soluzione scalare e modulabile in ambito Motion Logic e soluzioni PLC. Osservando il dettaglio di figura 7.1 si nota una struttura suddivisa a blocchi, partendo da sinistra si incontra:

- Un modulo per l'interfaccia con l'operatore, diretta attraverso un display e 4 pulsanti per navigare nei menu,e remota attraverso una porta ethernet con cui trasferire i programmi, quest'ultima utilizzata solamente per il dialogo da pc a PLC e non verso l'impianto. Appena sotto sono presenti il pulsante di reset e i Ready Contact, procedendo verso il basso troviamo uno slot per schede Compact Flash dove viene memorizzato il firmware e la parametrizzazione dell'impianto.
Proprio questa presenza è una caratteristica peculiare dei controllori e azionamenti Bosch Rexroth,

attraverso questa scheda di memoria è possibile trasferire le parametrizzazioni su impianti diversi o aggiornare i firmware dei dispositivi senza l'ausilio del pc.

-Al centro sono presenti due interfacce ethernet dedicate a SERCOS III (X7E1, e X7E2), una porta seriale RS485 con interfaccia PROFIBUS DP e altre due interfacce ethernet specifiche per reti Real Time Ethernet, quindi potenzialmente anche per EtherNet/IP. Non è stato possibile collegare questo controllore agli azionamenti presenti nell'altro impianto in quanto per utilizzare questa interfaccia è necessario acquistare dalla casa produttrice lo stack di comunicazione specifico della rete.

-A Sinistra trovano spazio e 8 canali di input e 8 di output ad alta velocità e infine l'alimentazione del dispositivo.

Fig. 7.1: IndraControl L65

Ogni dispositivo prodotto da Bosh Rexroth è identificato da un codice utilizzato sia per la parametrizzazione, sia per la definizione del prodotto con tutte le modifiche e opzioni montate. A tale scopo verrà di seguito illustrato e commentato il codice dei dispositivi utilizzati.
Il codice del controllore è CML65.1-3P-504-NA-NNNN-NW che va interpretato con il seguente significato:

- CM = Unità principale.
- L = Tipo di dispositivo in linea.
- 65 = Com Express.
- 1 = Utilizza un processore Intel Celeron da 1.0 GHz.
- 3P = equipaggiato con connessioni SERCOS III (3) e Profibus (P).
- 504 = sistema configurato con 256 MB di DRAM e 8 MB di SDRAM.

- NA = nessuna ventilazione.
- NNNN = nessun optional.
- NW = senza firmware a bordo.

Il dispositivo garantisce un livello di protezione IP20.
La versione Firmware caricata tramite scheda di memoria al momento del funzionamento è CML65s-MLs-04V22.0081.

Fig. 7.2: Driver SERCOS

I driver utilizzati visibili in figura 7.2 appartengono entrambi alla categoria advanced (offrono funzioni avanzate di controllo assi), ma differiscono per quanto riguarda la dotazione di interfacce e la presenza a bordo di un azionamento della tecnologia Safe On Board.

Il Driver 1 (a sinistra in figura) 7.2 ha codice: CSH01.1C-S3-ENS-EN2-MEM-S1-S-NN-FW che significa

- CSH = Modello CSH advanced
- 01 = Linea 1
- 1 = design 1
- C = configurabile
- S3 = comunicazione principale SERCOS III
- ENS = Opzione 1 Encoder IndraDyn/Hiperface /1Vpp/TTL
- EN2 = Opzione 2 Encoder EnDat 2.1/1Vpp/TTL
- MEM = encoder emulator
- S1 = Con tecnologia safety I/O
- S = pannello di controllo Standard
- NN = nessuna opzione aggiuntiva
- FW = indica che il firmware è indicato in un'altra posizione

Il firmware caricato è: FWA-INDRV*-MPH-05V16-D5-1-SNC-ML:

- FWA-INDRV* = IndraDrive firmware
- MPH prestazioni di livello ADVANCE
- 05V16 Versione 05
- D5 Indica le lingue supportate: Tedesco, inglese, francese, italiano e spagnolo
- 1 = abilitato ad operare in anello chiuso
- SNC = supporta la sincronizzazione elettronica
- ML = Con Motion Logic e funzioni tecnologiche

Mentre il Driver 2 (a destra in figura 7.2) ha codice CSH01.2C-S3-ENS-EN2-CCD-NN-S-NN-FW

- CSH = Modello CSH advanced
- 01 = linea 1
- 2 = design 2
- C = configurabile
- S3 = Comunicazione primaria SERCOS III

- ENS = Opzione 1 Encoder IndraDyn/Hiperface /1Vpp/TTL
- EN2 = Opzione 2 Encoder EnDat 2.1/1Vpp/TTL
- CCD = Abilitato alla comunicazione incrociata slave-to-slave
- NN = senza opzione safety
- S = pannello standard
- NN = nessuna altra opzione
- FW = indica che il firmware è indicato in un'altra posizione

Con firmware: FWA-INDRV*-MPH-05V16-D5-1-NNN-ML

- FWA-INDRV* = IndraDrive firmware.
- MPH prestazioni di livello ADVANCE.
- 05V16 Versione 05.
- D5 Indica le lingue supportate: Tedesco, inglese, francese, italiano e spagnolo.
- 1 = abilitato ad operare in anello chiuso.
- NNN = senza estensioni.
- ML = Con Motion Logic e funzioni tecnologiche.

Entrambi gli azionamenti sono dotati di uno slot per Compact flash dove memorizzare il firmware del dispositivo, tramite il modulo MultiMediaCard che permette all'utente di trasmettere o duplicare i parametri del driver compresi i dati sugli assi e l'orientato in maniera veloce e senza l'ausilio di un pc.
La scheda di supporto hardware è presente in due versioni:
- PFM02.1-016-NN-FW con drive firmware precaricato dall'azienda
- PFM02.1-016-NN-NW preformattata per il semplice trasferimento dei parametri da parte dell'utente

Entrambi i driver offrono le seguenti caratteristiche elettriche

Dati		
Corrente continuativa	A	11.3
Corrente massima	A	28.3
Potenza bus in continua con/senza choke	KW	5.1/5.1
Massima uscite senza/con choke	KW	8/10
Voltaggio nominale	V	3 AC 200 ... 500, 1 AC 200 ...0500 ($\pm$10%)
Corrente di ingresso nominale	A	13
Dipendenza dell'uscita dal voltaggio nominale		$U_{LN} < 400V$: riduzione di 1% della potenza ogni 4 V, $U_{LN} > 400V$: aumento di 1% della potenza ogni 5 V
Terminale per Bus DC		•
Capacità bus DC	μ F	270
Resistenza freno		
Resistenza freno		Interna
Massima energia dissipabile dal freno	kWs	5
Potenza continua di frenatura	KW	0.15
Potenza massima di frenatura	KW	10
Dati controllo tensione		
Controllo tensione interno	V	DC 24 (non per alimentare il mantenimento della frenatura)
Controllo tensione esterno	V	DC 24 $\pm$20% (DC 24$\pm$ 5% mantenendo frenato il motore)
Consumo di potenza senza unità di controllo e freno motore	W	14
Corrente continuativa senza unità di controllo e freno motore	A	0.6
Dati meccanici		
Spessore	mm	65
Altezza	mm	352
Profondità	mm	265
Peso	kg	3.8

Caratteristiche convertitore IndraDrive C - compact converters HCS02:

Oltre all'interfaccia principale SERCOS III utilizzata, i driver permettono di interfacciarsi verso altri dispositivi o reti:

Driver 1
Connettore x4: Interfaccia di comunicazione dedicata verso i motori.
Connettore x2: Interfaccia seriale su standard RS232
Connettore x41: Opzionale dedicato a funzioni di Safety tecnology.
Connettore X10: Interfaccia di emulazione encoder.
Connettore x8: Estensione degli I/O analogici.

Driver 2
Connettore x4: Interfaccia di comunicazione dedicata verso i motori.
Connettore x2: Interfaccia seriale su standard RS232
Connettori x24-x25: Interfacce ethernet SERCOS III utilizzate per le funzioni di Cross-communication tra slave.

Connettori x-26: Interfaccia ethernet Engineering Interface

Fig. 7.3: Motore sincrono MSK

I due motori sono entrambi di tipo sincrono e sono identificati dal codice :Msk060c-0600-nn-m1-ugo-nnnn

- Msk motori sincroni di tipo Msk.
- 060 = taglia.
- C = lunghezza totale.
- 0600 = tipo di canali.
- NN = raffreddamento a convezione naturale.
- M1 = Encoder Multigiro (Hiperface) 128 incrementi con 4096 rivoluzioni assolute.
- G = Albero liscio con anello di tenuta.
- 0 = Senza freno di stazionamento.
- N= Eccentricita albero standard, solo in abbinamento all'encoder S1 e M1.
- N = Altre dotazioni standard.

E offrono le seguenti caratteristiche elettriche e meccaniche:

Motore	velocità massima rpm	coppia conti-nuativa Nm	coppia massima NM	corrente nominale A	corrente massima A	momento d'inerzia kg m^2
MSK060 c-0600	6000 rpm	8 Nm	24 Nm	9.5	38.0	0,0008

7.2 l'ambiente di programmazione INDRAMOTION

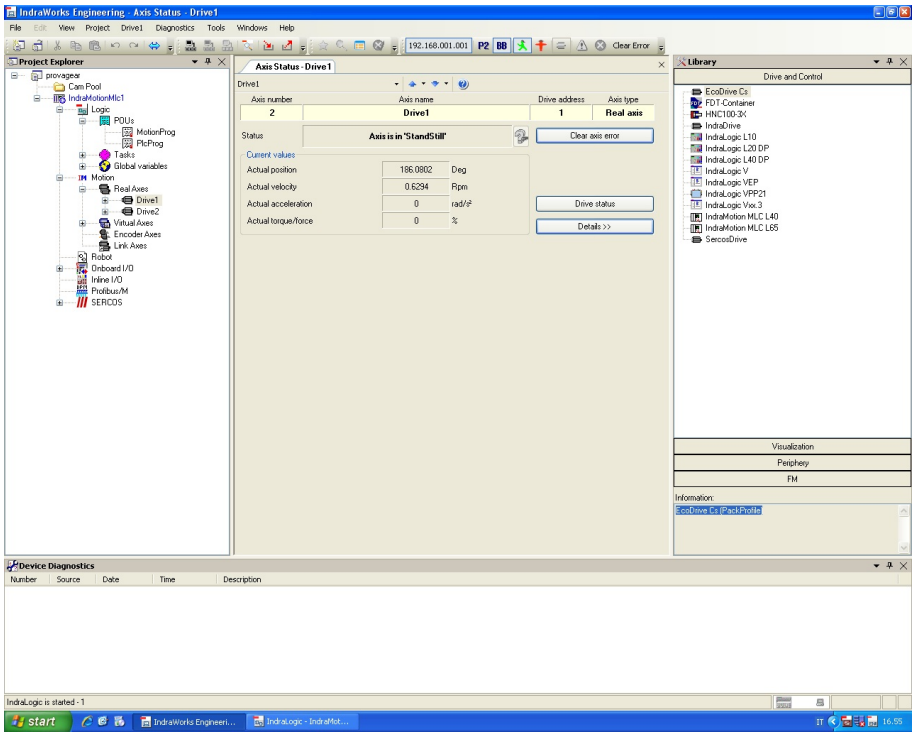


Fig. 7.4: L'ambiente di lavoro Indraworks

L'ambiente di lavoro offerto da Bosh Rexroth per gestire l'impianto e creare i programmi utilizzato è il software IndraWorks versione MLD 05V16.0050 SWA-IWORKS-MLD-05VRS-D0-CD650- Bosch Rexroth AG 2009.
Le funzionalità offerte da questo ambiente di lavoro sono decentrate su più applicativi diversi accessibili dalla schermata principale. Alcuni esempi sono: CamBuilder 04V11.0069, utilizzato per costruire il profilo delle camme elettroniche, IW-Drive 06V16.0119, utilizzato per gestire la parametrizzazione dell'impianto etc.
La schermata principale del programma visibile in figura 7.4 offre le funzioni di progettazione e parametrizzazione dell'impianto oltre che la diagnostica. Per ogni altro scopo è necessario richiamare un software differente, in particolare attraverso l'applicativo IndraLogic si richiama l'editor per i programmi e per l'interfaccia grafica del programma.
La scrittura del programma può avvenire tramite con il classico linguaggio ladder utilizzando la libreria precompilata offerta in bundle con il software, oppure in maniera testuale.
Mentre la progettazione del profilo di camma avviene nell'applicativo CamBuilder che offre sia modelli standard dedicati all'imbottigliamento e al rotary cut sia modelli liberamente configurabili in base alle proprie esigenze.
Una interessante funzione offerta da CamBuilder è la possibilità di importare, attraverso un comune file excel dei profili di camma generati su ambienti di terze parti, superando in questo modo eventuali limiti del programma. Sempre attraverso il programma IndraLogic è possibile creare delle interfacce grafiche (molto spartane) ai programmi, permettendo l'utilizzo del programma a personale non addestrato.

7.3 L'impianto EtherNet/IP

Fig. 7.5: L'impianto Allen Bradley

Il controllore utilizzato su questo impianto è il modello ControlLogix 5561^{TM}prodotto da Allen Bradley, numero di catalogo 1756-EN2T 1756 10/100Mbps Ethernet Bridge twisted pair media revisione 3.6 in grado di pilotare fino a 8 driver contemporaneamente per un totale di 128 assi, e un numero illimitato di assi controllati in coppia velocità o VHz. É compatibile con gli standard CIP Sync e CIP Motion.

Analizzando la struttura del controllore in figura 7.6 è possibile notare la suddivisione in moduli, partendo da sinistra sono presenti: il modulo di alimentazione , il controllore Logix 5561 il modulo dedicato alla comunicazione EtherNet/IP verso l'impianto, due moduli aggiuntivi di input e output, rispettivamente l'1756-IB32/B 32 Point 10V-31.2V DC Input e l'1756-OB32 32 Point 10V-31.2V DC Output e infine due moduli di interfaccia SERCOS II non utilizzati.
Il primo impatto rispetto alla controparte Bosch è la ridotta possibilità di interazione con l'utente, sul master sono presenti solamente 6 led di diagnostica del controllore e un interruttore a chiave per modificare lo stato di funzionamento tra Run, Rem (Remoto), Prog (programmazione). Nes-

Fig. 7.6: Logix 5561

suna opzione è stata dedica al dialogo diretto verso una persona fisica.
Il modulo di interfaccia EtherNet/IP presenta una sola porta ethernet, una porta USB di tipo B utilizzabile solo per la programmazione del dispositivo, e un display a scorrimento per la diagnostica di rete. La scelta di dotare il modulo di una sola interfaccia ethernet obbliga a utilizzare dispositivi aggiuntivi per creare topologie diverse dalla semplice linea.

Fig. 7.7: Kinetix 6500

I due azionamenti utilizzati, prodotti sempre da Allen Bradley, sono identici e formati entram-

bi da un modulo di controllo con identificativo: 2094-EN02D-M01-S0 Kinetix 6500 Single Axis Ethernet drive.
E una struttura portante modello 2094-BC01-MP5-M Kinetix 6500 che fornisce l'alimentazione per un controllo assi integrato (IAM) a 460V AC, 6 Kw, 4 A.
Ognuna delle due strutture così composte è dotata di 2 interfacce EtherNet/IP con "enbedded switch", integrano cioè le funzionalità di uno switch sulle due porte permettendo le connessioni ad anello e a linea aperta mantenendo le caratteristiche di rete switch ethernet.
Le performance offerte da questo azionamento sono:

- Anello di corrente a 1300 HZ.
- Anello di velocità a 500Hz.
- Controllo PWM a 8 KHz (BM01, BMP5) e 4 KHz (BM05-05)
- 250% Potenza di picco.

Sotto il profilo della sicurezza viene garantito il livello Cat.4 SIL3 con l'opzione di safe torque off. Questo garantisce che l'azionamento non eroghi coppia se in stato di stop.
Analizzando in dettaglio l'interfaccia notiamo subito che anche questi dispositivi non offrono la possibilità di interagire con l'utente, ma solo un display a scorrimento per la diagnostica base del driver. Dal punto di vista delle porte ausiliare è presente un solo connettore I/O, di sicurezza e di feedback ausiliario (IOD) a 44 pin.

Per quanto riguarda i motori, l'impianto monta due Brushless a bassa inerzia e con feedback assoluto, rispettivamente il modello MPL B420P-MJ72A e MPL B330P-J72AA.
Le loro caratteristiche sono riportate nella seguente tabella:

Numero catalogo	Velocità nominale (rpm)	potenza nominale (kw)	Inerzia rotore (kg-m^2)	Coppia di Stallo continua (Nm)	Coppia di stallo picco (Nm)	Corrente di stallo continua (A)	Corrente di stallo picco (A)
MPL-B420P	5000	1.9	0.00026	4.74	13.5	6.4	22.0
MPL-B330P	5000	1.8	0.00012	4.18	11.1	6.1	19.0

Questi motori contengono inoltre una memoria interna con le informazioni necessarie alla loro parametrizzazione, in modo che una volta connesso all'impianto il driver lo riconosca senza bisogno di intervenire manualmente.

7.4 L'ambiente di lavoro RSLogix 5000

La schermata principale del software RSLogix5000 integra tutte le funzioni necessarie alla gestione dell'impianto e alla programmazione del moto.
Attraverso il menu posizionato in alto a sinistra è possibile connettere il software con il master per parametrizzare l'impianto o trasferire un nuovo programma, mentre scendendo nei menu sottostanti troviamo il link all'editor dei programmi e il link allo stato delle variabili. Verso il fondo viene riprodotta la struttura dell'impianto con la possibilità di apportare modifiche e interrogare ogni singolo dispositivo per la diagnostica. Ogni azione apportata sulla colonna sinistra del programma richiama al centro dello schermo la funzione scelta senza cambi di struttura grafica o

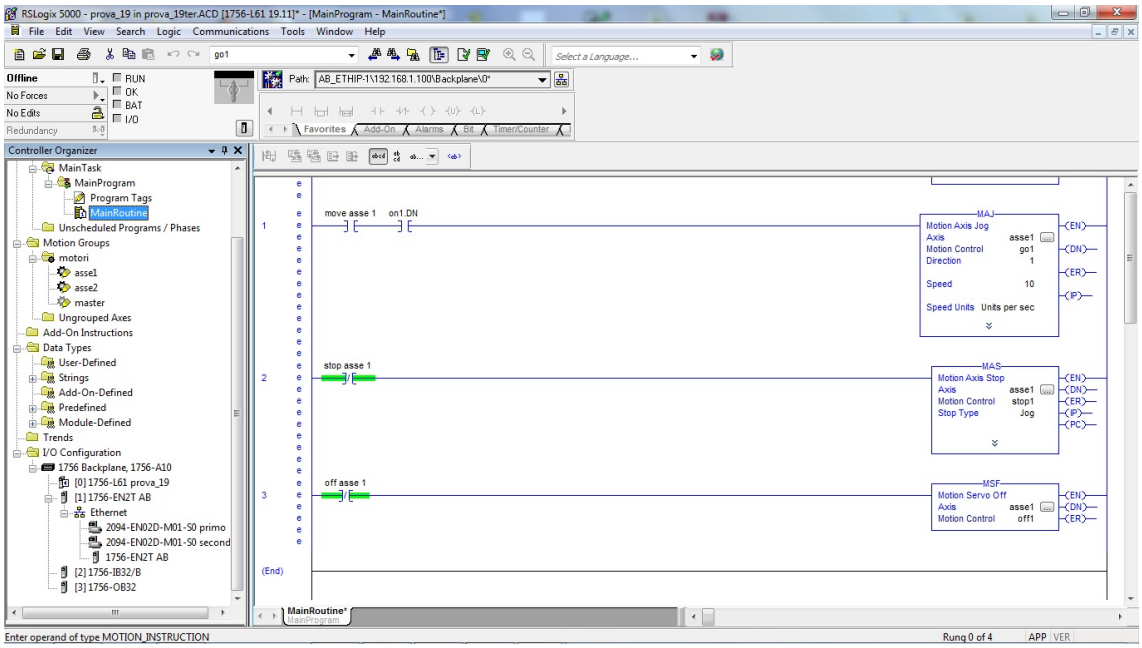


Fig. 7.8: L'ambiente di programmazione Allen Bradley

addirittura di software.
L'editor dei programmi è abbastanza similare a quello incontrato in ambiente Bosch, è basato sul linguaggio di programmazione ladder e sfrutta una libreria di oggetti standard creati da Allen Bradley. Le principali differenze risiedono nella impossibilità di dare una veste grafica ai programmi, e di conseguenza nel modo di interagire con essi.

7.5 Parametrizzazione degli impianti e scrittura programmi

Una volta montato l'impianto ed effettuate tutte le connessioni, prima di utilizzare l'impianto è necessario eseguire il commissioning, ovvero parametrizzare tutti i suoi componenti, verificare il funzionamento della rete e assicurarsi che tutte le funzionalità siano operative. Si intendono qui affrontare le problematiche riscontrate nel parametrizzare i due impianti e nello scrivere i programmi utilizzando i due ambienti di lavoro proprietari.

Parametrizzazione

La procedura di parametrizzazione è simile su entrambi gli impianti, si può sintetizzare in questi passi:

1. Per prima cosa si accende le rete, si attende che i dispositivi terminino la procedura di boot e si connette il pc al master.

2. Il passo successivo è parametrizzare il master della rete
   Su Allen Bradley è necessario accedere al programma RS linx classic Gateway, scegliere il menu Communication, configure driver, nella successiva schermata è possibile scorre la lista fino a trovare il controllore in nostro possesso e aggiungerlo alla rete con new. A questo

punto è visibile nella schermata principale del programma RS linx il dispositivo appena configurato, è bene definire immediatamente il suo indirizzo IP per velocizzare gli accessi successivi alla rete.
In maniera analoga al controllore, vengono configurati tutti i moduli aggiuntivi presenti nel master.
Su Bosch basta selezionare dalla colonna di destra il controllore IndraMotion MLC L65 e trascinarlo sotto la voce SERCOS III della colonna di sinistra. Nessuna altra configurazione è richiesta.

3. Successivamente si aggiungono i Driver.
   Su Allen Bradley si accede sempre al programma RS linx e attraverso il comando RSWho si cercano eventuali dispositivi connessi inn rete. Una volta individuati è necessario configurarli inserendo il codice (corretto !) del dispositivo e scegliendo il suo indirizzo IP.
   Su Bosch si trascina la voce IndraDrive sempre sotto SERCOS III, e si entra nel menu di parametrizzazione manuale del driver dove è necessario scegliere tutte le caratteristiche del dispositivo attraverso una serie di menu grafici. Un problema sorto in questa fase è stato la mancanza del motore utilizzato nei menu, si è risolto scegliendo un motore simile come caratteristiche e successivamente una volta instaurata la comunicazione con l'impianto scaricare la parametrizzazione corretta attraverso la funzione load default value.

4. A questo punto si è ricreato virtualmente l'impianto; è ora possibile accedere a tutti i dispositivi direttamente dalla schermata principale dei software Indraworks e RSLogix 5000 per modificare i parametri o eseguire diagnostiche.

In questa fase la rete Allen Bradley si è dimostrata migliore rispetto a Rexroth, infatti mentre la rete EtherNet/IP ha fatto immediatamente il ping dei dispositivi e in pochi minuti era operativa, la rete SERCOS III ha dato numerosi messaggi di errore che vanno dalla errata parametrizzazione degli I/O integrati nel controllore, a un generico messaggio riguardante le specifiche dei motori differenti da quanto indicato in fase di inizializzazione.
Inoltre mentre la parametrizzazione dell'impianto Allen Bradley è stata eseguita una sola volta, nell'impianto Bosch è necessario ripeterla ad ogni cambio di programma, perchè solo in questo modo è possibile rendere operativo l'impianto.

Scrittura ed esecuzione programmi

Entrambi i programmi utilizzano il linguaggio di programmazione ladder e offrono una completa libreria di funzioni. Sotto questo aspetto l'interfaccia grafica Allen Bradley fornisce una marcia in più, infatti le funzioni sono ordinate in categorie e selezionabili attraverso comodi menu a scorrimento, mentre in Bosch è necessario inserire un oggetto Box e premendo il tasto F2 scorrere la lista di tutte le funzioni finchè non si incontra quella desiderata.
Anche se il linguaggio di programmazione è lo stesso e i risultati ottenibili sono sostanzialmente gli stessi, la programmazione in Bosch è più simile a un diagramma a blocchi, dove le connessioni di ingressi e uscite determinano il comportamento dei blocchi. Invece Allen Bradley punta alla definizione testuale di tutte le variabili all'interno del blocco in modo da mantenere ordinato il programma.

La scrittura del programma in Indraworks ha dovuto mediare diverse esigenze, per prima cosa la scarsa disponibilità di software funzionante su cui basare il nuovo codice, poi la struttura molto completa e complessa della gestione delle rutine in Indraworks ha fatto ricadere la scelta sul mantenere una base comune della struttura modificando di volta in volta alcuni blocchi funzionali dei

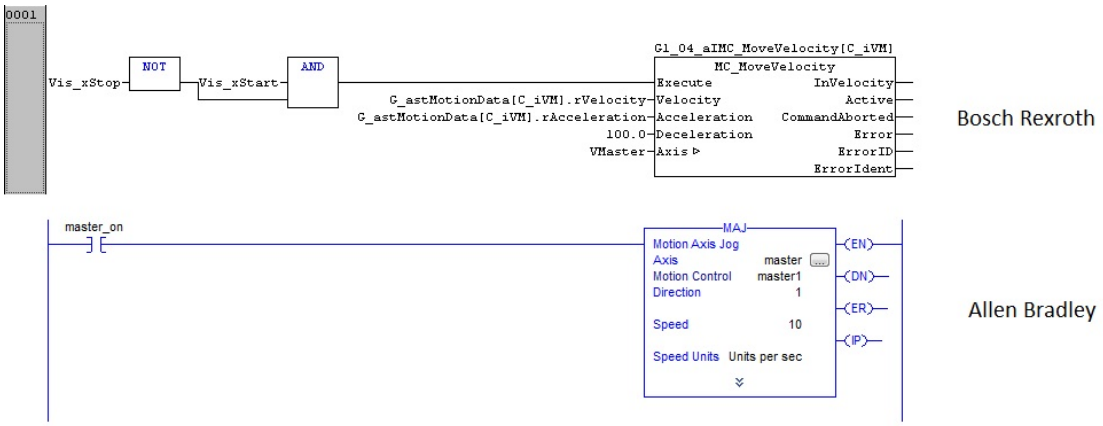


Fig. 7.9: Confronto delle interfacce di programmazione

vari programmi.

Il trasferimento del programma nel master e la sua messa in esecuzione è il passo che più ha creato problemi in Bosch, infatti ad ogni cambio di programma l'impianto "perdeva" la parametrizzazione e segnalava vari problemi risolvibili solo spegnendo e riaccendendo l'impianto. Un comportamento decisamente inaccettabile in ambito industriale che suppongo sia legato al particolare software o a un difetto nel master.

Capitolo 8

Test Motion Control

In questa sezione si affronta il difficile compito di creare dei programmi per i due impianti che permettano di valutare non solo le prestazioni degli impianti, ma anche delle reti di comunicazione.
Questo punto è stato molto complicato da affrontare, la diversità dell'hardware in particolare i motori e gli azionamenti è parso fin da subito un problema non risolvibile, inoltre i programmi devono essere "semplici" e facilmente replicabili su entrambi gli ambienti di lavoro, Indraworks e RSLogix 5000.
Questi programmi devono: generare nei due impianti del traffico "simile", con le informazioni di posizione "in chiaro", non devono fare ricorso a particolarità dell'impianto (come le camme trasferite nei driver degli azionamenti Bosch), e infine deve essere rappresentativo delle moderne problematiche dell'ambiente industriale.
La scelta è ricaduta su 2 filoni di programmi: il gearing e la camma elettronica.

Con gearing si intende la sincronizzazione in velocità o posizione di un asse rispetto ad un'altro. Questa funzione viene comunemente impiegata creando un asse virtuale definito master virtuale, al quale vengono poi legati attraverso opportune leggi, uno o più assi fisici reali. In questo modo una modifica sulla velocità del master si ripercuote sulla velocità di tutto l'impianto in maniera lineare e armonizzata.
Un'altra funzione tipica è legare due assi fisici che devono muoversi in assoluta sincronia, in questo modo il rallentamento dell'asse primario dovuto a cause ambientali provocherà un rallentamento dell'asse secondario, mantenendo pertanto la sincronia necessaria.

La camma elettronica riprende il concetto già noto in meccanica di camma, cioè lo spostamento secondo una legge particolare (profilo di camma) di un cedente rispetto al moto regolare e rotatorio di un movente.
Attraverso il controllo in posizione degli assi è possibile legare virtualmente ogni posizione dell'asse master (virtuale o reale che sia) a una precisa posizione degli assi slave permettendo di progettare gli movimento sincronizzati tra loro ma di forma diversa.

Questa decisione permette di generare varie tipologie di controllo di posizione, il più oneroso controllo possibile in entrambi gli impianti, permette un semplice confronto prestazionale degli impianti, e infine utilizzando velocità basse e spostamenti lunghi è possibile ridurre l'influenza delle differenze hardware.

Nelle due reti sono quindi stati eseguiti 4 test:

1. Controllo di posizione semplice.
   Utilizzando la funzione move absolute su Bosh e Motion Axis Move(MAM) su Allen Bradley, si comanda su entrambi gli assi uno spostamento di circa 180° a velocità costante e profilo di velocità trapezoidale.

2. Gearing tra asse virtuale e assi reali.
   Viene creato un asse virtuale e comandato in posizione con i blocchi visti nel punto 1. Attraverso la funzione Gear In Position di Bosch e Motion Axis Gear di Allen Bradley si legano entrambi gli assi reali a compiere i medesimi spostamenti dell'asse virtuale.

3. Gearing tra due assi reali.
   Viene comandato lo spostamento di un asse reale con le modalità viste del test 1, il secondo asse viene pilotato attraverso la funzione gearing vista al punto 2, con il riferimento di posizione proveniente dal primo asse.

4. Pentalatero.
   Sfruttando la struttura già montata sull'impianto Bosch visibile in figura 8.1 è stato progettato un profilo di camma per i due motori in modo che l'estremo libero del pentalatero eseguisse la figura di un cerchio.

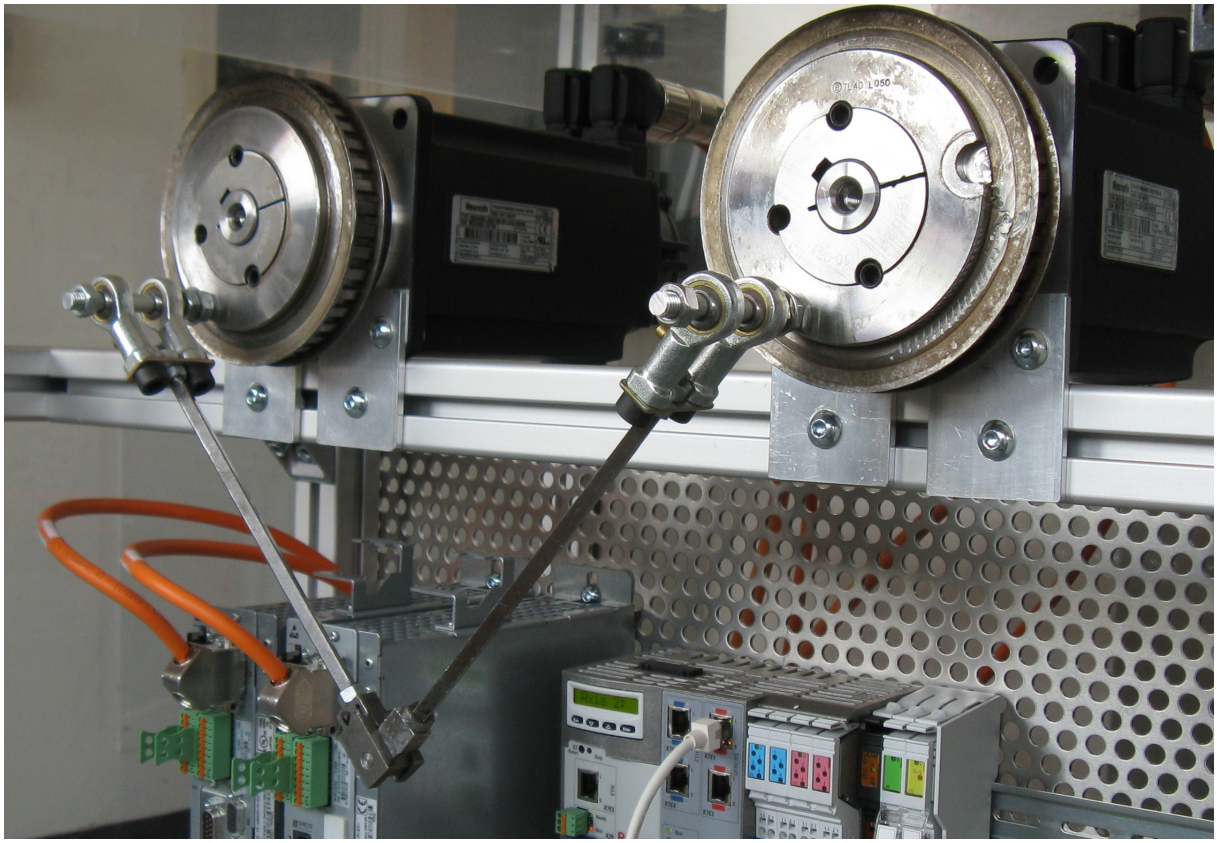


Fig. 8.1: Pentalatero montato sull'impianto

Per prima cosa è stata eseguita la cinematica inversa del pentalatero per legare la posizione del vertice del pentalatero con la posizione dei due motori. A tale scopo si è creata la seguente funzione in Matlab:

```
function [alpha,beta] = cinematicainversapentalaterobasso(xpp,ypp)

% Coordinate dell'end effector
xp=xpp;
yp=ypp;

% Lunghezza giunti
l1=45.0; % mm primo giunto motore a
l2=230.0; %mm secondo giunto motore a
l3=225.0; %mm secondo giunto motore b
l4=45.0; %mm primo giunto motore b
```

```
%coordinate albero motore a
xa=0;
ya=0;

%coordinate albero motore b
xb=300;
yb=0;

%Verifica dello spazio di lavoro
lap= sqrt((xp-xa)^2 +(yp-ya)^2);
lbp= sqrt((xp-xb)^2 +(yp-yb)^2);
if(lap>(l1+l2)||lbp>(l3+l4)||l2-l1>lap||l3-l4>lbp)
    error('errore, punto fuori dallo spazio di lavoro');
end

%calcolo angolo alpha motore a
costeta2a=(xp^2+yp^2-l1^2-l2^2)/(2*l1*l2);
senteta2a=-sqrt(1-costeta2a^2);
alpha = atan2(yp,xp)+atan2(l2*senteta2a,l1+l2*costeta2a);
% coordinate motore a

%calcolo angolo beta motore b
costeta2b=((xp-xb)^2+(yp-yb)^2-l3^2-l4^2)/(2*l3*l4);
senteta2b=sqrt(1-costeta2b^2);
beta =atan2((yp-yb),(xp-xb))+atan2(l3*senteta2b,l4+l3*costeta2b);
% coordinare motore b
```

Successivamente si è applicata questa cinematica a 1024 punti (dimensione massima delle camme sull'azionamento Bosch) equamente distribuiti sulla circonferenza del cerchio desiderato, esportando su file excel le posizioni relative dei motori.

```
% cerchio da descrivere con end effector
xc=155;
yc=-170;
raggio = 15;

% Lunghezza giunti
l1=45.0; % mm primo giunto motore a
l2=230.0; %mm secondo giunto motore a
l3=225.0; %mm secondo giunto motore b
l4=45.0; %mm primo giunto motore b

%coordinate albero motore a
xa=0;
ya=0;

%coordinate albero motore b
xb=300;
yb=0;

%costruzione cerchio
numeropunti = 1023; % numero di punti di cui è formato il cerchio
```

```
intervallopunti=2*pi/numeropunti;
t=0:intervallopunti:2*pi;
tdeg= radtodeg (t);
x=(raggio*cos(t))+xc;
y=(raggio*sin(t))+yc;
alpha = 0:intervallopunti:2*pi;
beta = 0:intervallopunti:2*pi;


%calcolo cinematica inversa sui punti del cerchio
for j=1:numeropunti+1
[alpha(j),beta(j)]=cinematicainversapentalaterobasso(x(j),y(j));
end
```

Verificando con simulazioni visibili in figura 8.2 la correttezza dei dati calcolati.

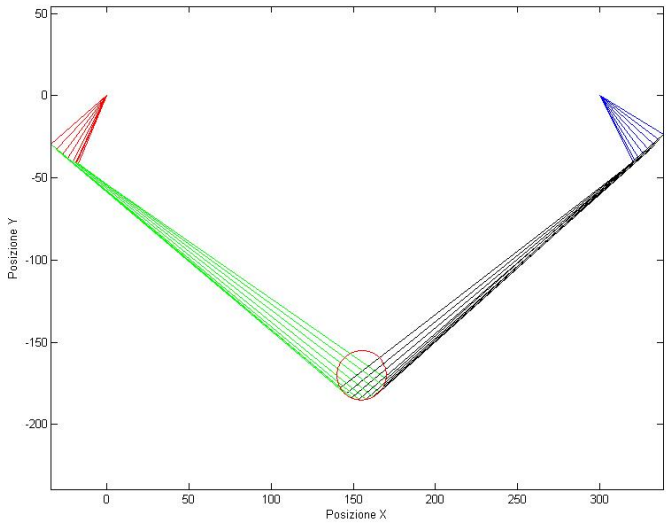


Fig. 8.2: Rappresentazione grafica della cinematica inversa

Successivamente i profili salvati su un file .CSV sono stati importati su software Cam Builder e trasferiti ai rispettivi driver.

Nel riproporre questo test sull'impianto Allen Bradley sono sorti alcuni problemi; la funzione "trasferisci la camma nel driver" è esclusiva degli azionamenti Bosch e quindi non è riproducibile su altri impianti, dove bisogna accontentarsi di generare la camma nel master e spedire ad ogni ciclo la posizione desiderata agli azionamenti.
Inoltre il software RSLogix 5000 non permette di importare camme da file esterni, quindi è stato necessario modificare la struttura del 4 test in modo da calcolare on-line il profilo di camma.
A tale proposito si è tradotto il programma Matlab per il calcolo della camma nel linguaggio di programmazione ladder, con le seguenti modifiche:

- Non viene riconosciuta la funzione $atan2(x,y)$, che viene sostituito dalla funzione $\pm acos(x/sqrt(x^2+y^2))$
- Il calcolo di 1024 punti per ciascuna camma è risultato troppo oneroso in termini di tempo da eseguire on-line, la funzione watch-dog del programma RSLogix 5000 mandava in blocco il controllore. Quindi si è ridotto il calcolo a 150 punti per camma

- Per ridurre ancora il carico di lavoro è stato diviso il calcolo dei due profili di camma in due programmi separati.

I programma così costruiti e visibili in appendice sono stati inseriti in due sub-rutine richiamate tramite il blocco Jump To Subroutine (JTS) nel corpo del programma.
Infine si è svolto un 5 test ripetibile su ambiente Bosch e Allen Bradley secondo lo stesso schema di comunicazioni.

5 Camma elettronica nel master:
Viene creata una sequenza di 4 posizioni che vengono ripetute ciclicamente per formare una traiettoria di più punti, queste posizioni vengono poi utilizzate per ricreare in entrambi gli assi una sequenza di spostamenti uguale.
Nello specifico:
-Sul'impianto Bosch la sequenza di posizioni è stata generata attraverso il blocco MUX a 4 ingressi, e comandata al master virtuale e ai motori con le stesse modalità del test 2.
-Su Allen Bradley sono stati utilizzati i blocchi standard per la camma: Motion calculate Cam Profile (MCCP)e Motion Axis Position Cam (MAPC).

Durante ogni test è stato analizzato il moto in tempo reale utilizzando l'oscilloscopio interno messo a disposizione dai due ambienti di lavoro Bosch e Allen Bradley. Contemporaneamente con Wireshark si è catturato il traffico circolante sulla rete in modo tale da analizzare la comunicazione e rappresentare graficamente il moto in maniera indipendente.

Proprio questa cattura dei pacchetti sulle due reti ha richiesto alcuni accorgimenti specifici:

- Su impianto Bosch la cattura è stata notevolmente facilitata dalla struttura logica della comunicazione, tutti i pacchetti circolanti in rete sono broadcast. É quindi bastato mettersi in ascolto nella porta libera dell'ultimo slave per catturare l'intero traffico presente in rete.

- La cattura dei pacchetti dell'impianto Allen Bradley è stata più complicata rispetto a Bosch, infatti la comunicazione si basa su messaggi trasmessi in unicast, inoltre ogni slave integra uno switch tra le sue due porte creando una rete switched.
  Attivando la modalità promiscua in Wireshark e andando a catturare il traffico sulla porta non utilizzata dell'ultimo slave si riesce ad osservare solamente i messaggi diretti dagli slave al master e non viceversa.
  Questo porta a concludere che i messaggi del master non attraversano tutta la rete, ma terminano il proprio viaggio quando vengono ricevuti dal destinatario. Mentre i messaggi spediti dagli slave transitano in tutta la rete.
  Per riuscire a catturare tutto il traffico è stato necessario cambiare il punto di ascolto sulla rete, o meglio creare un punto di ascolto tra la porta ethernet dell master e la porta ethernet del primo slave.
  Per ottenere questo sono state valutate diverse soluzioni:

  – HUB 100 Mbps
    Inserendo un HUB si ha la certezza che tutto il traffico che transita su una porta venga duplicato su tutte le altre. Questa soluzione ha il vantaggio che il prezzo dell'hardware è abbastanza basso (poco più di 10 euro), il problema maggiore di questa soluzione è il funzionamento solo in half-duplex del dispositivo che di fatto modifica le performance della rete.

- Switch con funzione di mirroring
  Utilizzando la particolare funzione diagnostica posseduta dagli switch di fascia alta è possibile replicare il funzionamento dell'HUB, con il vantaggio di mantenere la comunicazione in full duplex. Lo svantaggio è il prezzo molto elevato, la configurazione specifica per l'impianto in uso, i lunghi tempi di reperimento di questo dispositivo.
- Costruire manualmente un network-tap a partire da alcuni connettori ethernet.
  Questa soluzione ha un vantaggio economico rispetto alle prime due soluzioni, inoltre garantisce di non introdurre ritardi sulla comunicazione dovuti all'elettronica. Ma essendo un prototipo autocostruito non è possibile offrire garanzie sulle misure effettuate.

La scelta è ricaduta sulla prima soluzione principalmente per un discorso economico e di ripetibilità della misura.

8.1 Elaborazione e presentazione dei dati

Wireshark permette di catturare i dati, dissezionare i protocolli, tradurre i pacchetti in binario esadecimale e ascii etc. ma non consente di elaborare matematicamente e presentare in forma grafica i dati contenuti nei pacchetti.
Per fare ciò ci si è quindi dovuti affidare a un'altro software, Matlab, che con il suo ambiente di lavoro ha permesso di elaborare i dati acquisiti e presentarli sotto forma grafica.
Il principale problema incontrato tra Wireshark e Matlab è la totale assenza di applicazioni o toolbox che permettono di convertire i file catturati (.PcaP) in un formato compatibile con il software matematico. L'unica soluzione percorribile è stata quindi esportare i pacchetti in un formato temporaneo, ripulire il file dai frame superflui e infine importare in modo ordinato il file temporaneo in Matlab.
I passi principali di questa procedura sono stati:

1. Per prima cosa i file catturati in Wiresherk vengono filtrati in modo da eliminare tutto il traffico estraneo alla gestione degli azionamenti. Questo è stato fatto utilizzando i filtri: sercos type == AT in SERCOS e IP surce == 192.168.1.1 e IP surce ==198.168.1.2 in EtherNet/IP per selezionare solamente i pacchetti slave.

2. Una volta isolati, i pacchetti vengono esportati da wireshark in un file di tipo C Array utilizzando le impostazioni di default del programma.

3. I file ottenuti secondo questo procedimento contengono oltre ai byte del pacchetto anche delle informazioni testuali come la numerazione progressiva, la lunghezza, la traduzione in codice ascii etc. É quindi necessario "pulirli", a tale scopo sono stati importati in un foglio di calcolo excel e sfruttando la geometria del pacchetto e la funzione trova e sostituisci si sono rimossi gli elementi indesiderati. Salvando infine il tutto in file di testo (.TXT)

4. Infine viene utilizzata la funzione *textscan* per importare il file di testo in Matlab. Il codice utilizzato a tale scopo è:

```
%%%%%% Procedura di importazione pacchetti %%%%
fid = fopen('pacchetto.txt','r');
if(fid>0)
vet=textscan(fid,'%s',2999999);
```

```
%vet1=textscan(fid,'%s',1249595,'delimiter','\n');
end
fclose(fid);
```

Ottenendo una vettore colonna dove a ogni cella corrisponde un solo byte del pacchetto.

5. A questo punto conoscendo la posizione dell'informazione desiderata all'interno del pacchetto e la lunghezza totale del pacchetto è stato possibile scorrere tutto l'array estraendo solo le informazioni scelte:

```
%%%%% estrazione dati dai pacchetti catturati %%%%%
nummaster = 10000; %numero pacchetti da analizzare
pos1=[nummaster];
poaa=[nummaster];
indiceposizione1= 123; % posizione del primo dato nel
% primo pacchetto

for i=1:nummaster %ciclo for che scandisce i pacchetti

%Di seguito vengono estratti i 4 byte che formano
%l'informazione, vengono messi nel giusto ordine
% e infine convertiti in decimale
primo1= vet{1,1}{indiceposizione1+3,1};
hex1= hex2dec(primo1);
if (hex1 <= 9)
primo1 =strcat('0',primo1);
end
secondo1=vet{1,1}{indiceposizione1+2,1};
hex2= hex2dec(secondo1);
if (hex2 <= 9)
secondo1 =strcat('0',secondo1);
end
terzo1 =vet{1,1}{indiceposizione1+1,1};
hex3= hex2dec(terzo1);
if (hex3 <= 9)
terzo1 =strcat('0',terzo1);
end
quarto1 =vet{1,1}{indiceposizione1,1};
hex4= hex2dec(quarto1);
if (hex4 <= 9)
quarto1 =strcat('0',quarto1);
end
intermedio1=strcat(primo1,secondo1,terzo1,quarto1);
posa=hex2dec(intermedio1);
if (i==1)
    zero = posa;
end
%attraverso questa procedura si estrae l'offset
%tra posizione zero e numero di impulsi-giro associati

%Conversione da encoder a posizione
posb = posa-zero;
```

```
impgiro = 2097152;
pos1(i)= posb*(360/impgiro)
indiceposizione1=indiceposizione1+126;
% in questo modo si passa alla medesima posizione
% del pacchetto successivo
end
% a fine ciclo for l'array pos1 contiene
% l'elenco ordinato delle informazioni estratte dai pacchetti
```

La procedura sopra descritta, con le opportune modifiche in base a dimensioni del pacchetto e al numero di dati da estrarre, permette di importare le informazioni contenute nei pacchetti in un array matlab.
Parallelamente all'importazione avviene anche la conversione dei dati in gradi, che circola in formato diverso nei due impianti:

- In SERCOS III i dati di posizione presenti nei pacchetti vanno interpretati leggendo il dato in byte procedendo da destra verso sinistra, convertendo infine da esadecimale a decimale e scalando di un fattore 1000. Il dato così ottenuto è direttamente una posizione compresa tra 0° e 360°.
  Questo perchè la gestione dell'encoder assoluto e la conversione tra impulsi dell'encoder e posizione del motore avviene nel driver, che quindi può trasferire al master direttamente l'informazione finale.

- In EtherNet/IP i dati vengono sempre letti a singolo byte da destra verso sinistra e convertiti in esadecimale, ma il numero che si viene a formare non rappresenta una posizione in gradi, ma un totale di impulsi encoder.
  É quindi stato necessario calcolare un opportuno coefficiente di conversione a partire dalla risoluzione dell'encoder (2097152 impulsi/giro) e memorizzare a ogni test eseguito l'offset tra zero macchina e l'equivalente zero nella misura dell'encoder.
  Questo comportamento è dovuto al fatto che gli encoder assoluti vengono gestiti dal controllore centrale in questo impianto Allen Bradley, quindi i dispositivi comunicano tra loro in termini di impulsi giro e solamente nel controllore questi numeri acquisiscono il loro reale significato.

Il secondo problema affrontato è l'ordine dei pacchetti. Wireshark infatti cattura tutte le comunicazioni presenti sulla rete secondo l'ordine di arrivo al punto di ascolto.
I pacchetti che trasportano le informazioni dell'impianto si trovavano quindi inseriti in maniera non ordinata all'interno del traffico catturato. É stato quindi necessario trovare un sistema che permettesse di ordinare i pacchetti in modo da sincronizzare le informazioni appartenenti allo stesso ciclo di comunicazione. Per fare ciò è stato indispensabile affidarsi all'analisi delle due reti fatta nel capitolo 6.

- Su rete SERCOS III, l'informazione del ciclo a cui appartengono i dati, è trasportata in maniera implicita nella comunicazione. Infatti sul mezzo trasmissivo si alternano un pacchetto MDT e un pacchetto AT. Quest'ultimo contiene i feedback di entrambi gli azionamenti.
  Quindi per eseguire un confronto ordinato è sufficiente accoppiare tra loro i dati presenti nello stesso pacchetto AT.
  Per quanto riguarda l'informazione temporale dei dati, il tempo di ciclo dell'impianto è di 250 μs e quindi le informazioni vengono catturate e presentate a tale distanza.

- Su EtherNet/IP invece l'ordine di arrivo dei pacchetti non segue generalmente schemi prestabiliti, ma analizzando le catture svolte, si nota invece il seguente ordine dei pacchetti:

Master$->$Slave1, Master$->$Slave2, Slave1$->$Master, Slave2$->$Master Master$->$Slave1 etc... Questo accade probabilmente per le ridotte dimensioni della rete, e per l'assenza di switch o altre infrastrutture intelligenti.
Inoltre la prima parte del campo dati CIP è un contatore di cicli che incrementa il suo valore solamente nel passare da un ciclo al successivo.
Per eseguire un confronto ordinato è quindi sufficiente accoppiare tra loro i dati contenuti nei messaggi con lo stesso conteggio di ciclo.
Su rete EtherNet/IP invece l'informazione temporale è trasportata nel pacchetto secondo il protocollo di CIP Sync, ed è presente un apposito campo dati che identifica il momento in cui il dato è stato acquisito sull'impianto.

In questo modo è stato possibile confrontare tra loro le posizioni degli assi in modo congruo anche senza l'utilizzo della costosa scheda di rete con time-stamp dell'ordine dei ηs.

8.2 Risultati sperimentali su SERCOS III

Si propone ora in maniera ordinata i risultati ottenuti sull'impianto Bosch Rexroth. Ricordando che sono stati svolti in condizioni di impianto ottimali, utilizzando cioè due motori uguali, disposti secondo il medesimo orientamento spaziale, pilotati da due azionamenti con caratteristiche elettriche identiche. Questo ha permesso di ottenere una omogeneità dei dati e una ripetibilità delle misure elevate.
Durante tutti i test il tempo di ciclo della rete è stato impostato in maniera automatica dal software Indraworks a 250 μs

8.2.1 1° TEST: controllo di posizione

Nel primo test viene programmato l'impianto per eseguire uno spostamento simultaneo dei due motori. Nello stesso istante gli assi partono dalla posizione 360° e raggiungono i 180°, non è presente nessun ulteriore meccanismo di sincronizzazione tra i due movimenti.

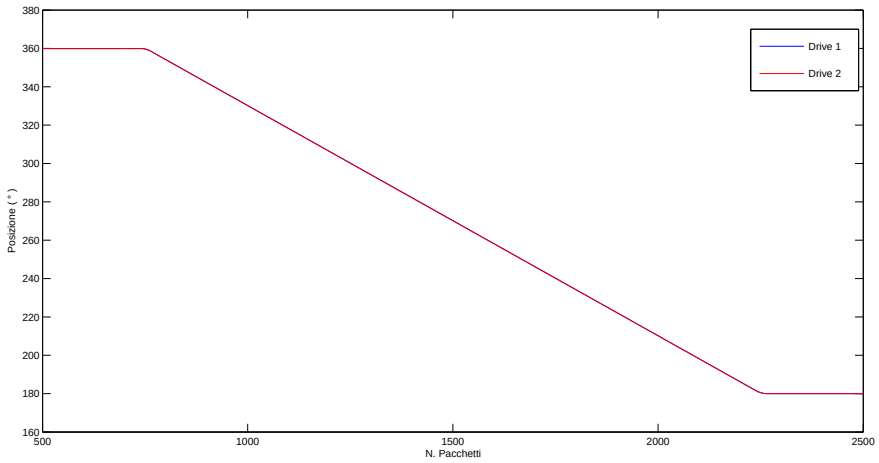


Fig. 8.3: Controllo di posizione

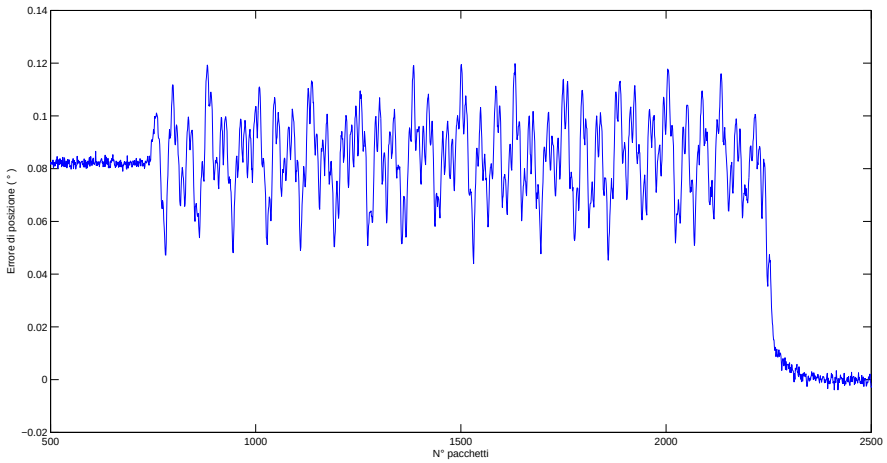


Fig. 8.4: Errore di sincronia in controllo di posizione

Come visibile in fig 8.3 e 8.4, il non aver sincronizzato tra loro gli assi, fa si che uno scostamento di circa 0.08° presente all'inizio della movimentazione perduri fino al raggiungimento del target di posizione.
Una volta raggiunto l'obiettivo questo offset si annulla, ma solo in conseguenza al fatto che i due motori raggiungono la medesima posizione.
É infatti visibile in figura 8.4 che il livello di precisione con cui entrambi gli assi raggiungere una posizione target è dell'ordine dei 0.002°.

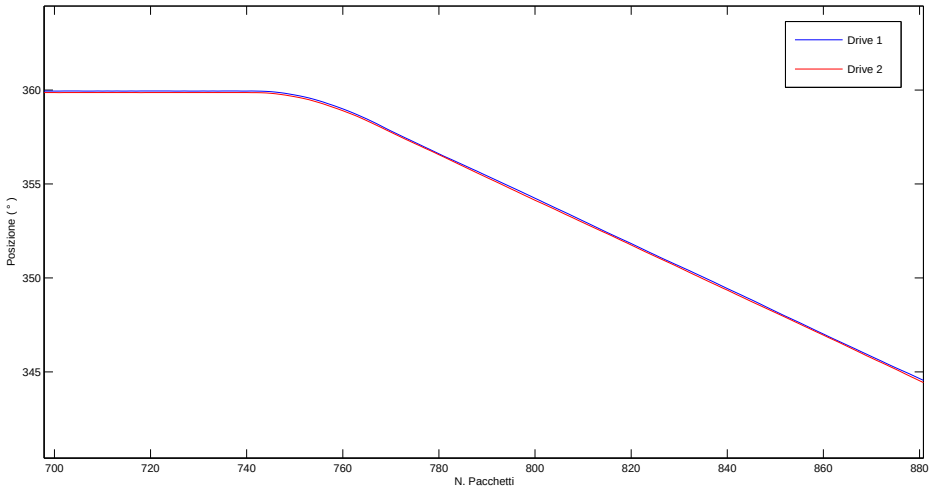


Fig. 8.5: Dettaglio del controllo di posizione

Nel dettaglio di Figura 8.5 risulta chiaramente cosa comporta in termini di traiettoria del moto un offset costante.

8.2.2 2° TEST: Gearing tra master virtuale e slave reali

In questo test, visibile in figura 8.6, si mette in atto una prima strategia di sincronizzazione del moto. Viene creato un asse master virtuale e ne viene programmato il moto da 0° a 180°, successivamente sempre nello stesso programma, si lega lo spostamento dei due assi reali allo spostamento del master.
Si trasmette in questo modo la stessa traiettoria nei due assi fisici, perche entrambi seguono il medesimo riferimento del master.

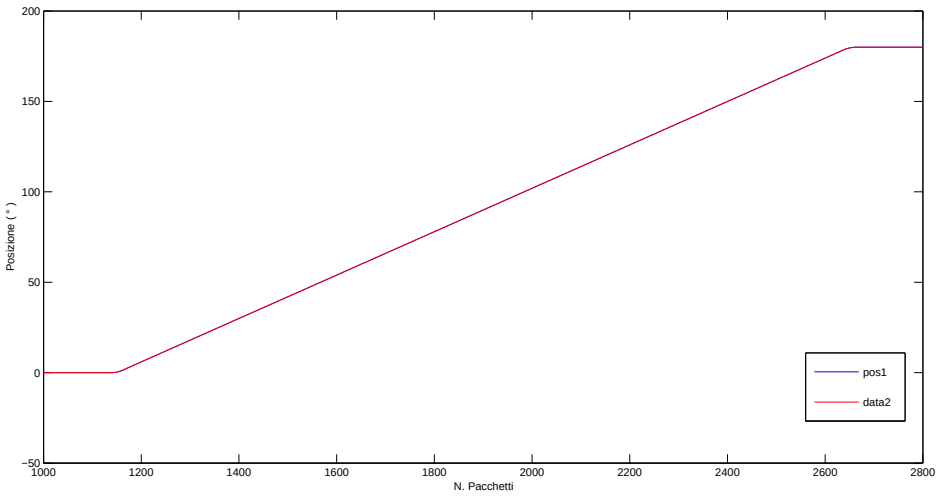


Fig. 8.6: Controllo con gearing sul master

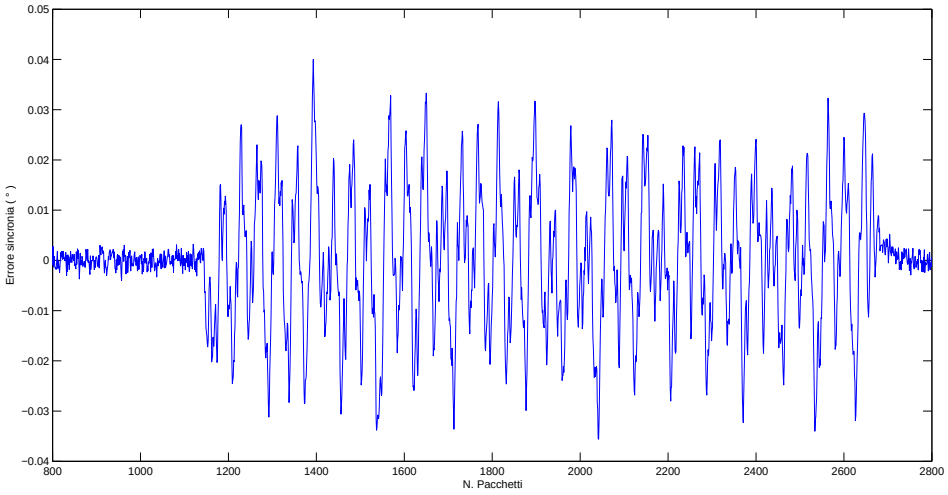


Fig. 8.7: Errore di sincronia in controllo gearing master

Come visibile in figura 8.7, gli offset tra i due assi sono scomparsi, questo è il piu evidente effetto del gearing. Inoltre durante lo spostamento l'errore di sincronia tra i motori è simmetrico rispetto lo zero, quindi il controllo di sincronizzazione bilancia gli spostamenti cercando di mantenere un valore medio nullo.
Il livello di sincronizzazione ottenuto con questo test può essere così riassunto:

- valore medio dell'errore assoluto 0.0120°
- errore massimo $\pm$ 0.04°

8.2.3 3° TEST: Gearing tra due assi reali

In questo test viene ripetuto il medesimo movimento del test 2 ma si utilizza una strategia di sincronizzazione differente. Viene programmato un comando diretto di posizione per l'asse 1, e contemporaneamente si utilizza sempre l'asse 1 come riferimento da inseguire con l'asse 2.
In questo caso una eventuale rallentamento dell'asse 1 dovuto a cause ambientali e non previsto nella pianificazione del moto, ha immediate ripercussioni sull'asse 2.
Questa strategia di controllo obbliga la rete a far transitare il riferimento di posizione prodotto dall'asse 1 due volte; una da slave 1 a master e l'altra da master a slave 2. É interessante notare come reagisce in termini di sincronizzazione l'impianto in seguito a questo ritardo.

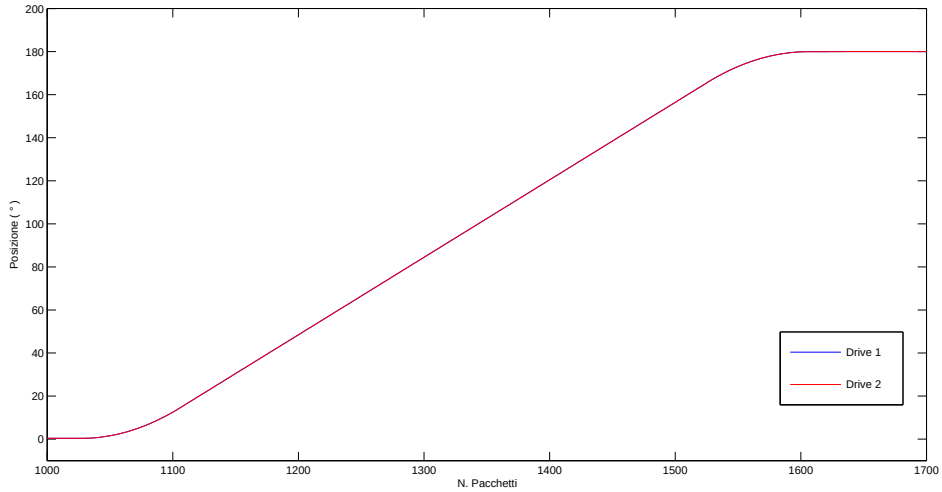


Fig. 8.8: Controllo di posizione con gearing tra slave

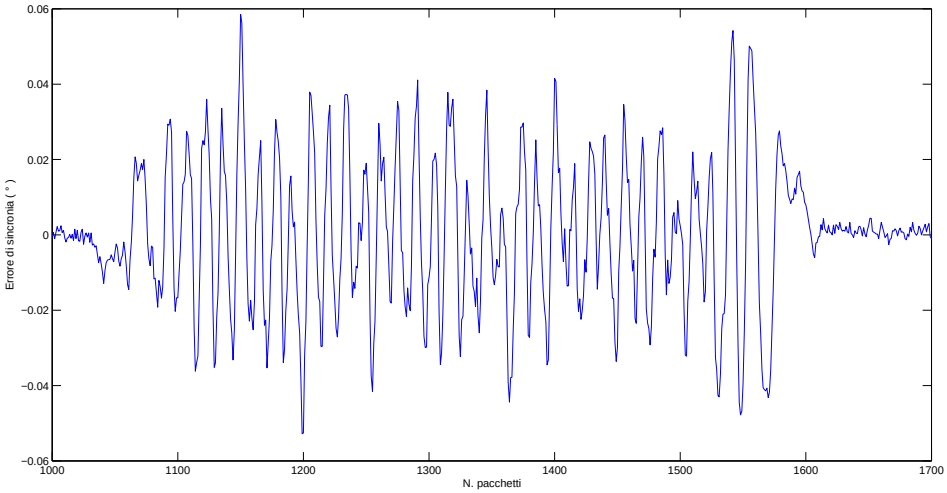


Fig. 8.9: Errore di sincronia in controllo gearing slave

Come visibile in figura 8.8 e 8.10, il livello di sincronia subisce un leggero calo:

- media dell'errore assoluto 0.018°
- errore massimo $\pm$ 0.058°

Tra il test 2 e 3 rimane invariato sia la traiettoria da compiere con i due motori, sia la configurazione hardware; quindi il calo di prestazioni è interamente da attribuire al ritardo con cui

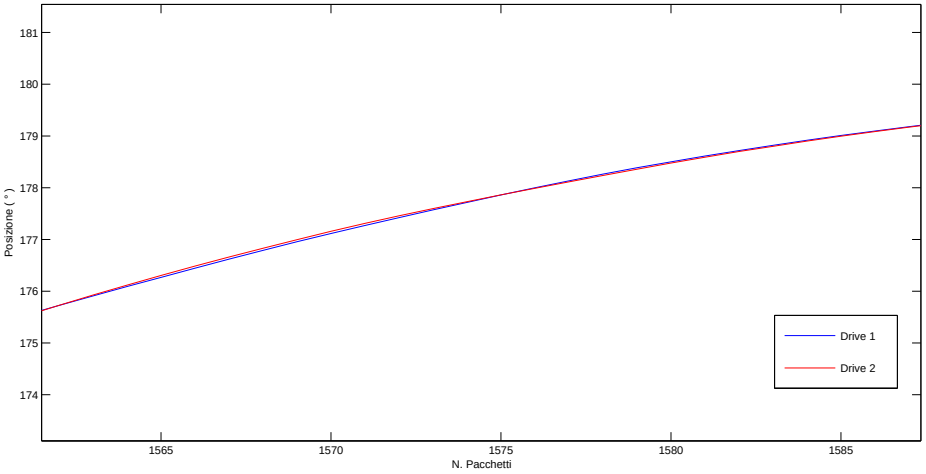


Fig. 8.10: Dettaglio del controllo di posizione con gearing tra slave

l'asse 2 riceve il riferimento di posizione. Un incremento dell'errore che comunque non modifica sostanzialmente la bontà della sincronizzazione come visibile in figura 8.10.

8.2.4 4° TEST: pentalatero

Per lo svolgimento di questo test viene montato sull'impianto la struttura del pentalatero, come visibile in figura 8.1, e programmato il moto reciproco dei due motori in modo da descrivere la figura del cerchio con l'estremo libero.
Questo test ha dato risultati pratici particolarmente soddisfacenti sull'impianto Bosch, ma è anche il meno significativo per quanto riguarda lo studio della rete. Questo perche si è sfruttato una funzione speciale degli azionamenti Bosch, il trasferimento del profilo di camma negli slave; in questo modo il master non trasmette direttamente agli azionamenti la posizione di riferimento, ma solamente l'incremento da dare alla camma precaricata nella memoria del driver. Cosi facendo la rete risulta sgravata da tutto il traffico feedback che si chiude nell'azionamento e non piu nel master.
Pur consci di questo fatto si è svolta lo stesso la prova per comprendere le massime prestazioni ottenibili dagli azionamenti.

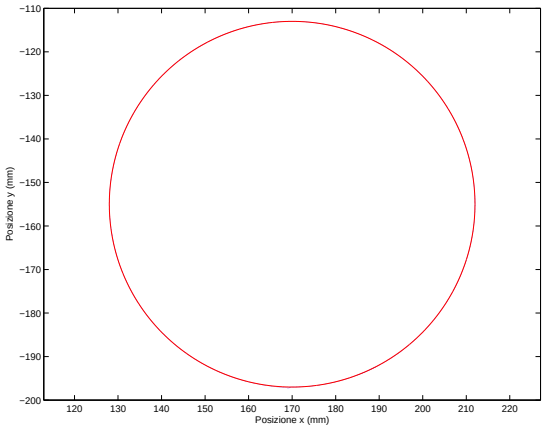


Fig. 8.11: Pentalatero

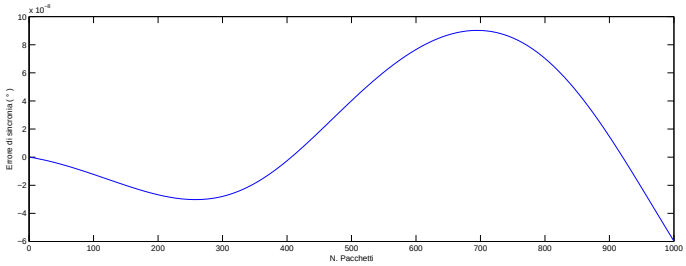


Fig. 8.12: Errore sulla traiettoria di alpha

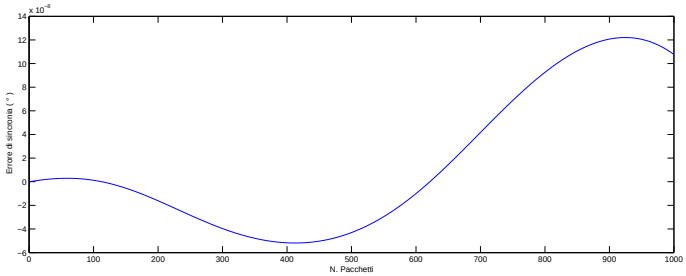


Fig. 8.13: Errore sulla traiettoria di beta

In figura 8.11 viene riportato in rosso il cerchio da ottenere e in blu la cinematica diretta delle posizioni misurate sui due motori, le due traiettorie sono praticamente sovrapposte. E' significativo osservare anche gli andamenti degli errori di sincronia tra la camma caricata nel driver e la posizione assunta dai motori.
Come era logico aspettarsi, l'errore visibile in figura 8.12, 8.13 è di diversi ordini di grandezza inferiore rispetto a quanto ottenibile con il controllo sul master.

8.2.5 5° TEST: camma elettronica

A scopo puramente dimostrativo per poter eseguire un confronto tra l'impianto Bosch e l'impianto Allen Bradley viene progettata una nuova camma piu semplice, la cui traiettoria è formata da 4 punti: 360°, 180°, 90°, 270° percorsi ciclicamente da entrambi i motori.
Questa camma viene calcolata nel controllore centrale e trasmessa ai due motori utilizzando un master virtuale e due gearing tra master e ciascun asse reale (lo stesso procedimento del test 2).

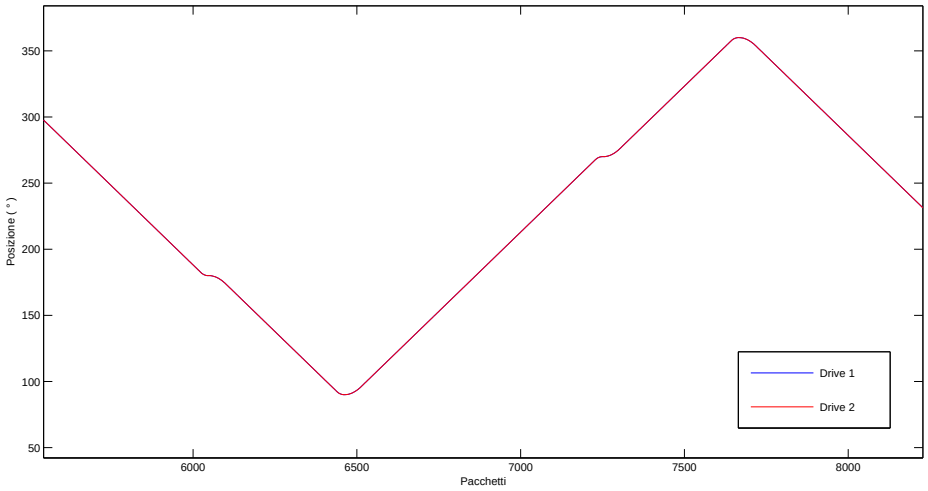


Fig. 8.14: Camma su master Bosch

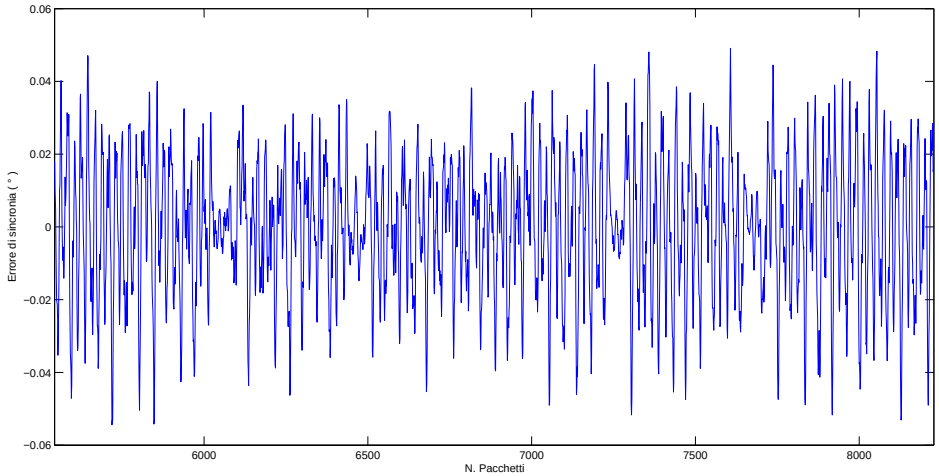


Fig. 8.15: Errore di sincronia in camma

Come visibile in figura 8.14 e 8.16 la precisione peggiora leggermente rispetto al test 2. Questo potrebbe sorprendere dato che si utilizzano le stesse strutture di controllo per eseguire lo sposta-

mento, in realtà bisogna tenere presente che i motori in questo caso non partono da condizioni di riposo, ma da una situazione di moto, quindi l'inerzia, l'inversione del moto etc. influenza chiaramente le prestazioni.
É invece interessante notare come all'avvicinarsi della posizione di riferimento si riduca l'errore di sincronizzazione, visibile confrontando l'andamento di figura 8.14 con 8.15, questa è un'altro esempio di come la velocità dei motori influenza la precisione di sincronizzazione reciproca.

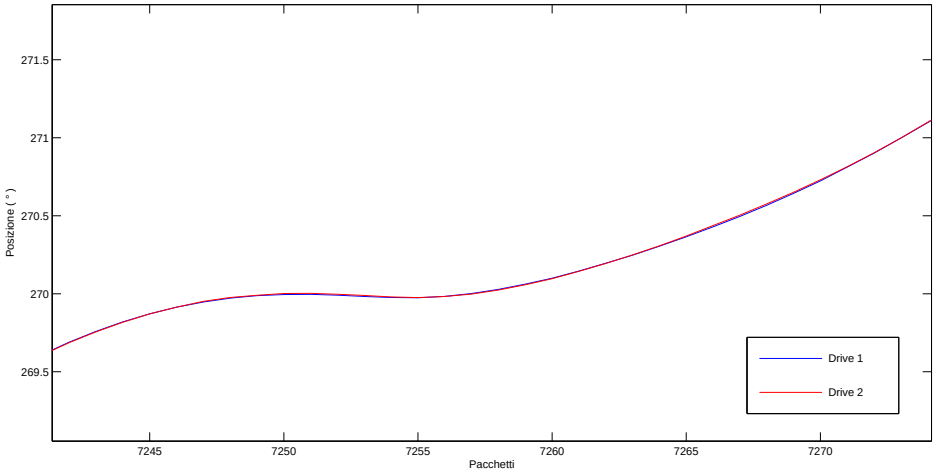


Fig. 8.16: Dettaglio camma su master Bosch

8.3 Risultati sperimentali su EtherNet/IP

I test condotti su questo impianto si sono svolti utilizzando un tempo di ciclo per la rete di comunicazione di 2000 μs nominali, calcolato cioè mediando tra di tutti i tempi di aggiornamento dell'impianto.
In particolare grazie al software RSLogix 5000 e alle sue funzioni di analisi e debug della rete EtherNet/IP, è possibile studiare in maniera più approfondita questo aspetto, andando ad analizzare le statistiche prodotte dalla rete:

	tempo medio μs	tempo massimo μs	nominale μs
master$->$slave	115	447	666
slave$->$master	120	7246	666
Jitter	360 ηs	1865 ηs	

Da dove si comprende che la criticità maggiore nei tempi si ha nella comunicazione tra slave e master.

La seconda importante differenza è la gestione dei messaggi, ogni slave comunica con il master attraverso un proprio messaggio, e il master risponde singolarmente a ogni slave. Questo genera una quantità di traffico in rete circa doppia rispetto all'impianto Bosch per trasportare le stesse informazioni.

La struttura del messaggio non è ovviamente formata da identificatori come in SERCOS, e non è stato possibile recuperare informazioni riguardo al pacchetto trasmesso in rete dal software RSLogix 5000. Quindi l'unica soluzione percorribile è stata eseguire dei test per individuare almeno l'informazione della posizione degli azionamenti.

Prima di passare all'analisi prestazionale della rete ci si è posto il problema di quanto l'introduzione del HUB influisca sulle prestazioni complessive dell'impianto.
Si è quindi eseguito il test più gravoso per la rete, il gearing tra i due slave, e confrontate tra loro le due prove: la prima eseguita senza HUB, prelevando i pacchetti dallo slave in fondo della rete con il risultato visibile in figura 8.24 la seconda inserendo l'HUB in rete come descritto e prelevando i pacchetti da una delle sue porte, figura 8.18.

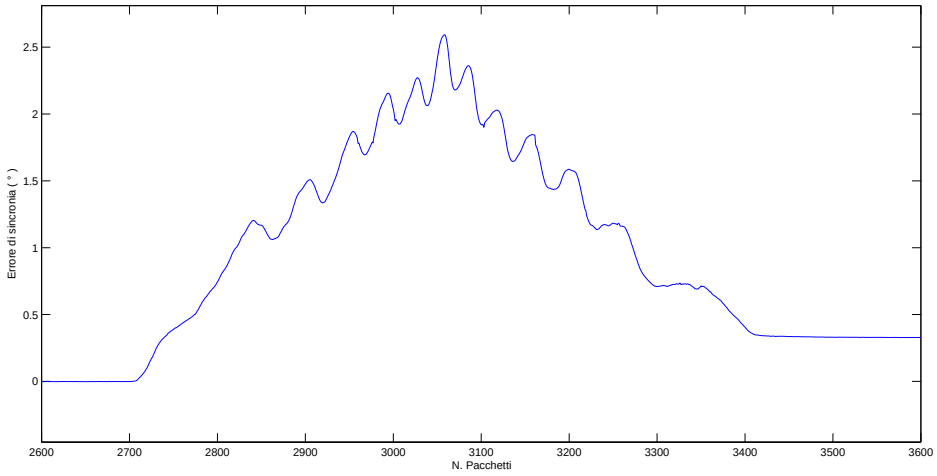


Fig. 8.17: Errore in gearing Full-Duplex

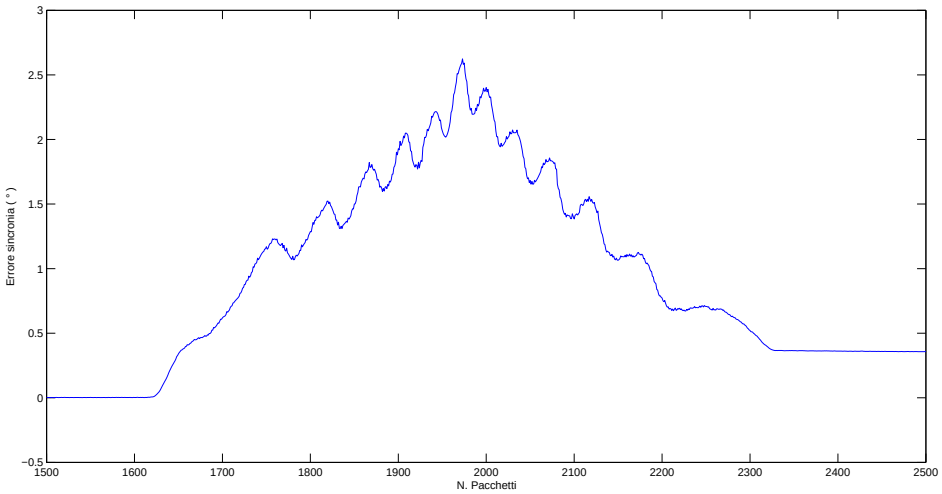


Fig. 8.18: Errore in gearing Half-Duplex

Confrontando i due test si notano varie cose:

- Lo scostamento medio e massimo tra i due assi è uguale.
- Il numero di pacchetti ricevuti dal master durante lo spostamento resta pressochè invariato.
- La misura in half duplex mostra sovrapposto all'errore normale del'azionamento un altro rumore di frequenza più elevata, indice che la comunicazione ha subito variazioni.
- La forma e l'andamento dell'errore risultano sostanzialmente invariati tra i due test.

Queste osservazioni permettono di concludere che l'inserimento dell'HUB non ha influenzato in maniera macroscopica le misure, che quindi possono ancora essere utilizzate in maniera comparativa.

8.3.1 1° TEST: Controllo di posizione

In questo test, visibile in figura 8.19, viene programmato lo spostamento simultaneo da 0° a 180° dei due motori senza nessuna strategia di sincronizzazione del moto.

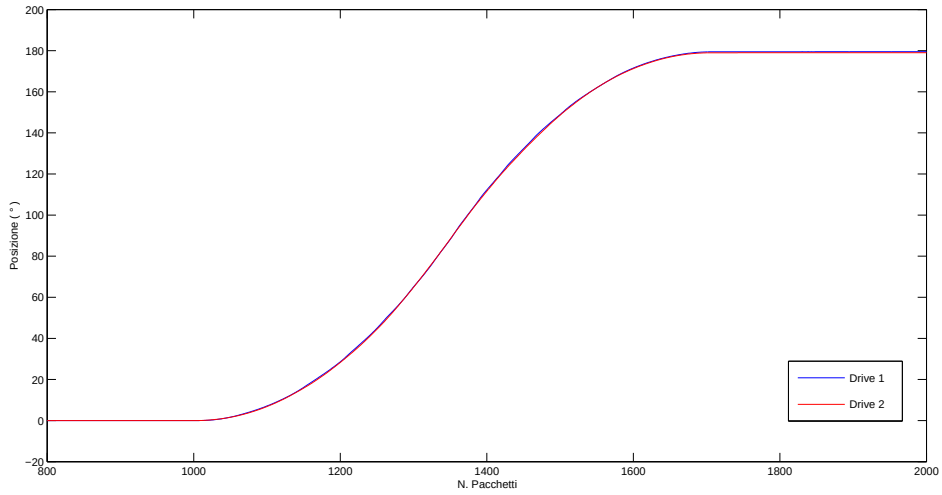


Fig. 8.19: Controllo di posizione su impianto Allen Bradley

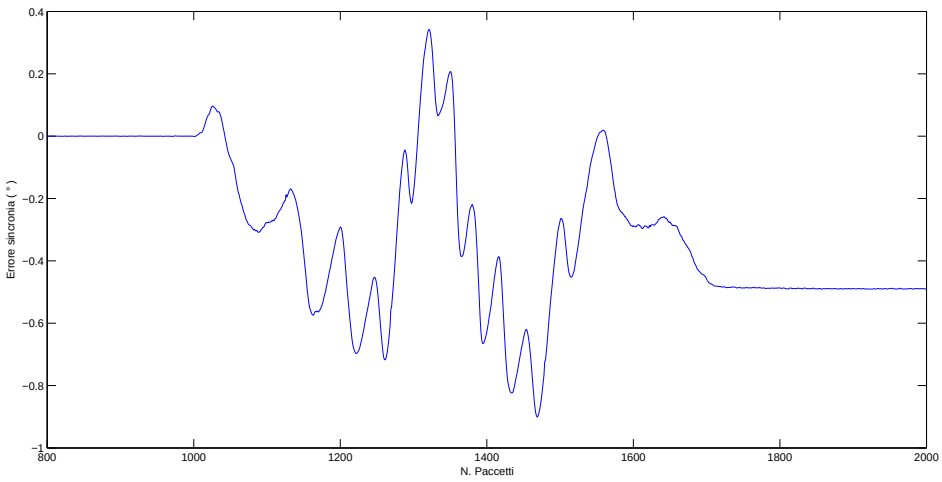


Fig. 8.20: Errore di sincronia con controllo di posizione

Osservando la figura 8.4 si nota la presenza di un offset sulla posizione reciproca dagli slave come nel medesimo test condotto sulla precedente rete,.
Si nota inoltre che l'andamento dell'errore ha caratteristiche completamente diverse rispetto all'impianto Bosch, è meno nervoso e ha uno schema di comportamento tipico; cresce nella prima metà del moto, tende ad annullarsi circa a metà dello spostamento, per poi tornare ad aumentare nella seconda metà del moto, oltre al fatto che non è a media nulla. Questo profilo visibile piu marcatamente nel test 2 indica delle lacune non solo sulla velocità di sincronizzazione della rete, ma anche nella strategia di controllo dei motori.

8.3.2 2° TEST: Gearing tra master virtuale e slave reali

In questo test, figura 8.21 si utilizza la strategia di sincronizzazione basata su un master virtuale che contemporaneamente esegue uno spostamento e genera il riferimento di posizione per sincronizzare i due slave.

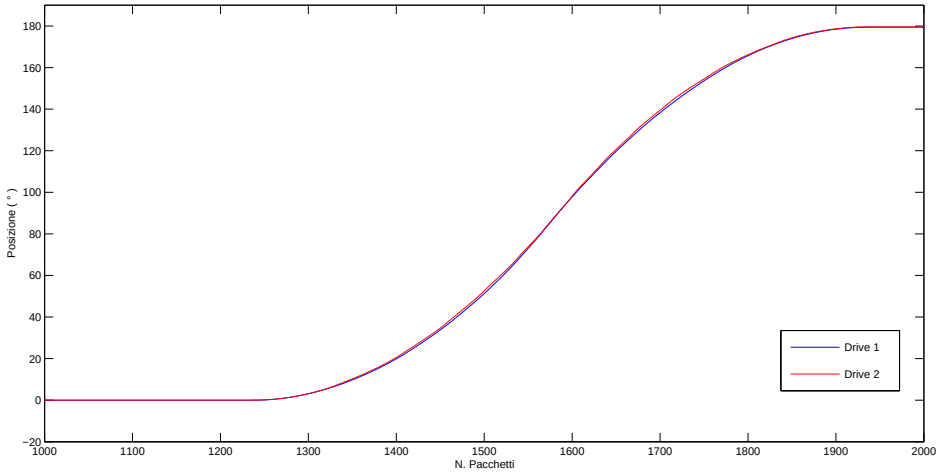


Fig. 8.21: Controllo in gearing master su impianto Allen Bradley

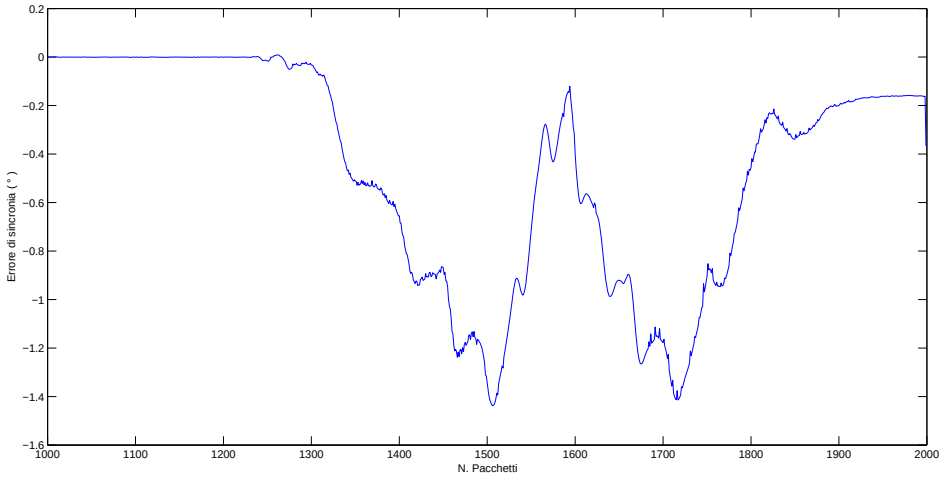


Fig. 8.22: Errore di sincronia con controllo gearing master

Dalla figura 8.22 si nota una forte riduzione dell'offset rispetto al semplice controllo di posizione. Riduzione ma non scomparsa, è presente infatti uno scostamento di circa 0.16° anche alla fine del moto. Questo offset sarà una presenza costante anche nei successivi test eseguiti su questa rete, la sua origine però è da attribuire maggiormente alla differenza di hardware nei due azionamenti piuttosto che alle prestazioni della rete di comunicazione.
Per quanto riguarda l'errore durante il moto si può riassumere con:

- Media dell'errore assoluto 0.6455°
- Errore massimo 1.43°

8.3.3 3° TEST: Gearing tra due assi reali

I questo test, figura 8.23, si utilizza la strategia di sincronizzazione basata solamente su assi reali. Viene quindi programmato lo spostamento diretto dell'asse slave1 e contemporaneamente all'asse slave2 viene spedito il riferimento di posizione prodotto dalla traiettoria dell'asse slave1.

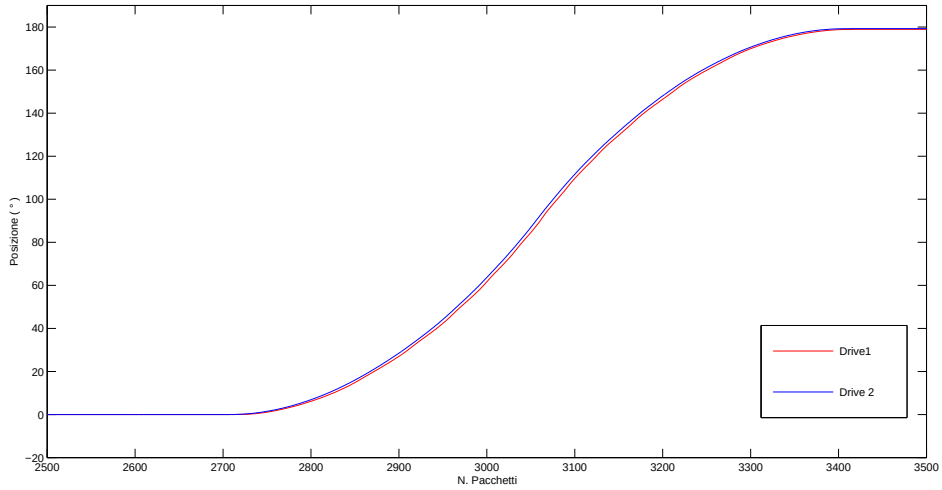


Fig. 8.23: Controllo in gearing tra due assi reali in Allen Bradley

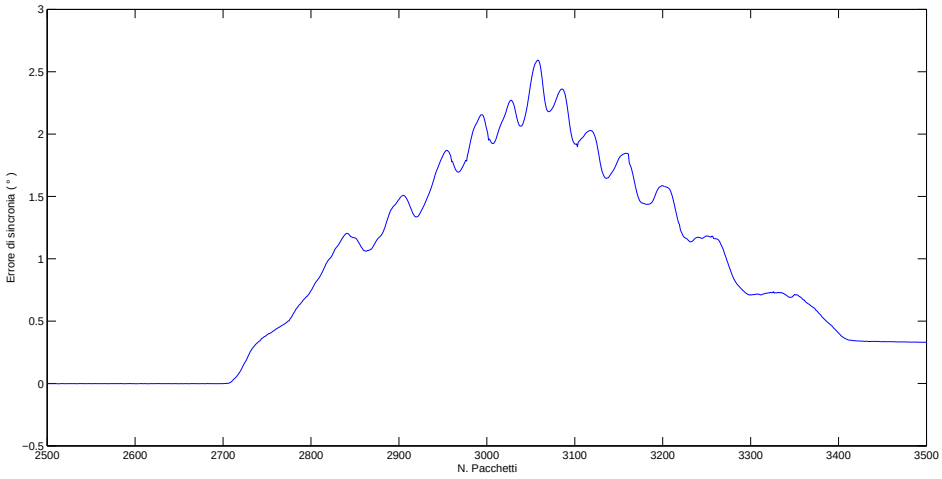


Fig. 8.24: Errore in gearing Full-Duplex

L'errore visibile in figura 8.24, è aumentato notevolmente, come l'offset presente alla fine del moto.

- Media dell'errore assoluto 1.2686°
- Errore massimo 2.59°

Anche in questo caso valgono i discorsi fatti precedentemente per l'impianto Bosch, il calo di prestazioni complessivo è da attribuire solamente alla rete e al suo comportamento in questo particolare test. Che evidentemente mette in luce i suoi limiti maggiori.

8.3.4 4° TEST: pentalatero

In questo test viene utilizzata la funzione camma per eseguire la figura del cerchio con il pentalatero.
Attraverso una sub-rutine nel programma principale si esegue il calcolo della cinematica inversa del pentalatero su un totale di 150 punti per ciascun asse, punti che successivamente il master trasmetterà in rete ai due driver.

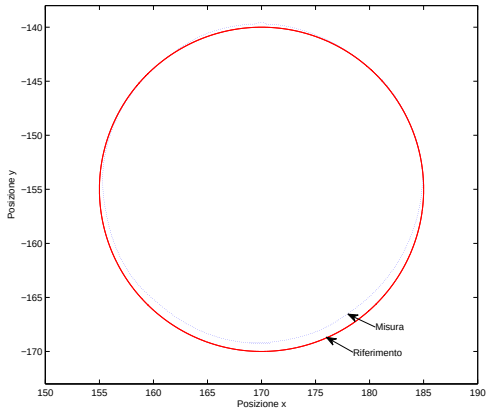


Fig. 8.25: Pentalatero su impianto Allen Bradley

I risultati visibili in figura 8.25, sono decisamente deludenti, il cerchio non è centrato e nella parte bassa si discosta di oltre 1 mm dal riferimento. Questo peggioramento rispetto allo stesso test condotto su Bosch non è da imputare solamente al minor numero di punti utilizzati nella cinematica inversa, ma anche e soprattutto all'errore commesso dai driver nell'inseguire il profilo di camma.

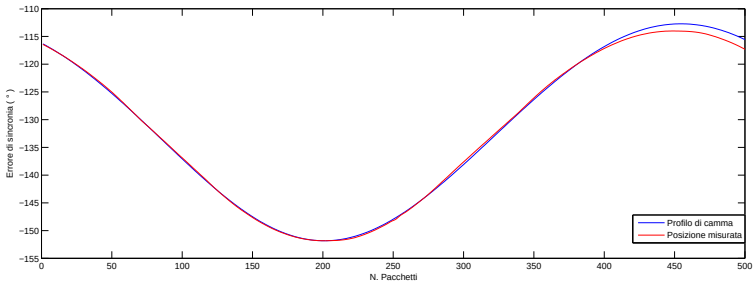


Fig. 8.26: Errore sul profilo di camma alpha

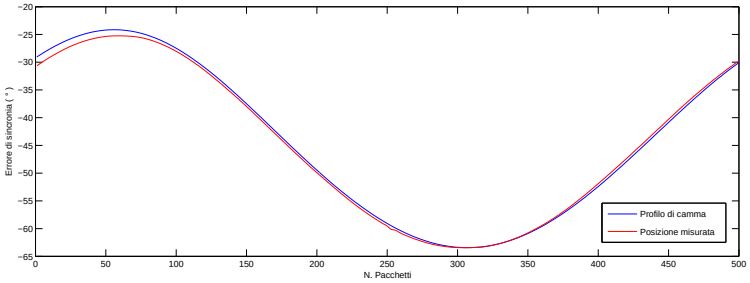


Fig. 8.27: Errore sul profilo di camma beta

Analizzando lo scostamento tra riferimento e traiettoria reale, visibile in figura 8.26 e 8.27 si nota che è di poco inferiore ai due gradi, coerente con quanto trovato nei test precedenti. La particolarità di questa applicazione pratica è che piccoli errori di inseguimento nei due assi vengono amplificati dalla struttura del pentalatero e generano marcati scostamenti dal cerchio di riferimento.

8.3.5 5° TEST: camma elettronica

In questo ultimo test viene riproposta una versione semplificata della funzione camma, viene programmato il moto dei due motori attraverso 4 posizioni; 0°, 180° 90° 270°.
Come si nota in figura 8.28 l'asse 2 è sempre piu lento dell'asse 1 ad inseguire la posizione.

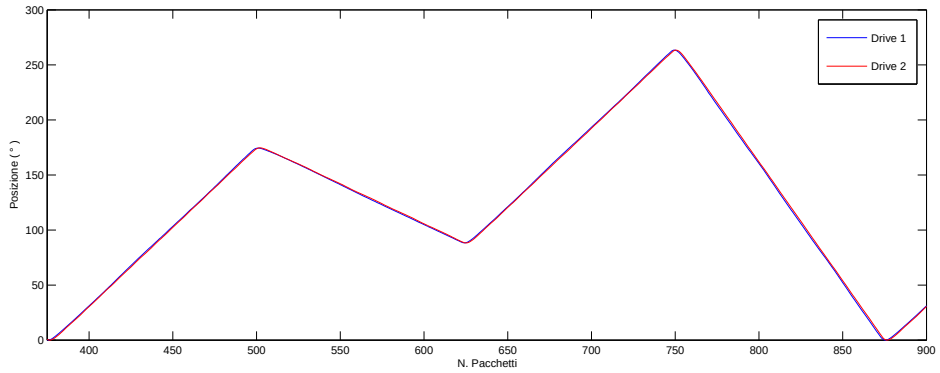


Fig. 8.28: Camma elettronica su Allen Bradley

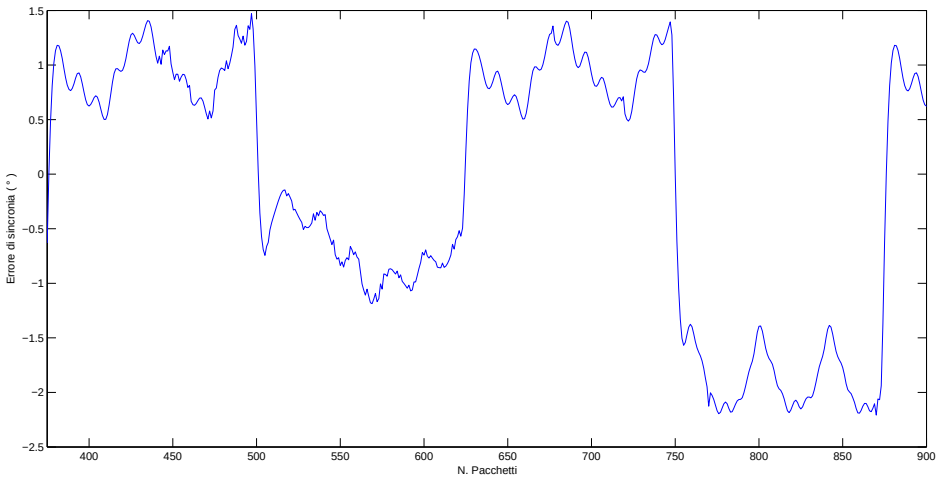


Fig. 8.29: Errore in camma Allen Bradley

Analizzando nello specifico l'errore, figura 8.29, si nota che la sua ampiezza è in relazione allo spostamento effettuato, è compreso tra 1° e 1.5° con spostamenti minori o uguali di 180°. ma aumenta oltre i 2° quando si esegue lo spostamento finale di 270°. Questo fa supporre un certo legame tra errore e entità dello spostamento.

Conclusioni

Le due reti di comunicazione sono state analizzate nei modi possibili compatibilmente con l'hardware a disposizione, le informazioni acquisite hanno permesso di comprendere la struttura della comunicazione, la forma dei pacchetti, la gestione logica dei messaggi e come vengono gestiti i riferimenti di posizione.
Lavorando con i software proprietari Bosch Rexroth e Allen Bradley è stato inoltre analizzato come avviene la parametrizzazione di un impianto e la scrittura dei programmi di Motion Control sulle differenti piattaforme di lavoro.
Nello svolgere la tesi è stato dedicato maggior tempo alla rete SERCOS III su impianto Bosch Rexroth in quanto l'unica operativa e funzionante al momento dell'inizio della tesi. Successivamente, nell'ultimo mese e mezzo, è stato possibile analizzare anche la rete Ethernet/IP su impianto Allen Bradley e fare alcuni confronti tra i due.

La valutazione degli ambienti di lavoro RSLogix 5000 e Indraworks è stata negativamente influenzata dal comportamento anomalo di quest'ultimo, messaggi di errore casuali, parametrizzazione impossibile di alcune funzioni, lunghi tempi di messa in servizio sono state solo alcune delle inspiegabili difficoltà incontrate nell'utilizzo di questo impianto.

Le analisi sperimentali hanno delineato un quadro abbastanza chiaro sulle prestazioni raggiungibili con i due impianti:
La rete SERCOS III su impianto Bosch Rexroth ha fatto registrare in tutti i test eseguiti le prestazioni migliori in Motion Control, per contro la costruzione dei programmi è stata più laboriosa e ha impiegato un tempo molto maggiore rispetto all'impianto Allen Bradley.
La rete EtherNet/IP con funzioni di CIP Motion ha dato risultati leggermente inferiori in campo Motion Control. Il motivo di questo calo prestazionale è di difficile individuazione, in parte influisce la velocità di comunicazione inferiore, il tempo di ciclo aumenta di circa 10 volte rispetto a SERCOS III, ma anche motori, driver e controllore hanno la loro parte di responsabilità. Il budgeting di queste cause richiederebbe un'analisi approfondita dell'hardware utilizzato, cosa che esula dall'obbiettivo di questa tesi che resta l'analisi della struttura della comunicazione. Mentre la programmazione e la gestione dell'ambiente di lavoro è stato decisamente migliore, la parametrizzazione è stata eseguita una sola volta e non sono stati mai riscontrati problemi di stabilità del software.

I problemi principali incontrati nello svolgimento della tesi sono tutti legati alla mancanza di hardware specifico:

- Scheda di rete con time-stamp del frame ethernet dell'ordine del μs come le schede netAnalyzer di hilsher
- Uno switch industriale con porta di mirroring per eseguire le misure in full-duplex sulla rete EtherNet/IP.
- La possibilità di avere un controllore e due driver che supportassero entrambe le interfacce EtherNet/IP e SERCOS III, per confrontare la comunicazione a parità di hardware.

Gli sviluppi futuri proposti in questo ambito sono un'analisi delle comunicazioni di tipo temporale, possibile solo utilizzando l'hardware sopra consigliato. Creare nuovi test di Motion control e utilizzare velocità maggiori per i motori, in modo da rendere ancora piu evidente gli eventuali ritardi introdotti. E come ultimo punto estendere questo studio a nuove reti come Profinet o EtherCAT.

Ringraziamenti

Ringrazio innanzitutto i miei genitori, perchè senza il loro costante aiuto e incoraggiamento non avrei potuto portare a termine gli studi.
Ringrazio poi il mio relatore Giovanni Boschetti per la fiducia posta in questo mio progetto di tesi, per la pazienza con cui ha sopportato le mie continue richieste di colloquio, e infine per i suggerimenti su come risolvere i problemi sorti
Ringrazio inoltre Il professore Dario Richiedei e il tecnico di laboratorio Nicola de Rossi per i loro insegnamenti su come utilizzare l'impianto Bosch-Rexroth.

Bibliografia

Per la stesura della tesi sono stati consultati i seguenti libri, articoli, e documentazione divulgazioni on-line:

-Industrial Communication Technology Handbook, Autore Richard Zurawski, CRC Press 2005.

Per lo studio della rete SERCOS III è stato consultato il sito di SERCOS:
-http://www.sercos.com/technology/sercos3.htm

In particolare nell'analisi degli identificativi IDNs è stato di vitale importanza il documento:
Rexroth IndraDrive Drive Controllers MPx-02; MPx-03; MPx-04.
Che contiene una completo ed esaustivo elenco degli identificatori comunemente utilizzati.

Mentre per la rete EtherNet/IP si è consultato il sito di ODVA:
-http://www.odva.org/Home/ODVATECHNOLOGIES/EtherNetIP/tabid/67/lng/en-US/Default.aspx
e
-http://www.odva.org/Home/ODVATECHNOLOGIES/EtherNetIP/EtherNetIPLibrary/tabid/76/lng/en-US/Default.aspx

Le informazioni sul materiale hardware degli impianti sono state reperite sul sito Bosch Rexroth e Allen Bradley :
-http://www.boschrexroth.com/dcc/Vornavigation/ -http://literature.rockwellautomation.com/idc/groups/public/do

Per l'analisi delle prestazioni massime raggiungibili dalle reti ci si è basati sulla pubblicazione:

-Industrial ethernet Technologies: Overview

Prodotta da EtherCAT Technology Group nell'agosto 2011.

Appendice A

Cavi e connettori per industrial ethernet

Il livello fisico di ethernet è ufficialmente descritto dallo standard IEEE 802.3, che ne fissa le caratteristiche principali.
Gli standard IEEE 802.3 utilizzati in industria sono:

- Su cavo in rame

10 Mbps Base-T
100 Mbps base-TX
1000 Mbps Base-T

- Su fibra ottica

10 Mbps Base-FL
100 Mbps Base-FX
1000 Mbps Base-SX

Esiste anche uno standard ethernet su cavo coassiale, ma non viene in genere utilizzato.

La scelta del supporto fisico influenza la massima distanza raggiungibile con un singolo collegamento, la velocità massima raggiungibile e il livello di reiezione ai disturbi ottenibile.
Con il cavo CAT5e la massima distanza percorribile con una singola tratta, senza infrastrutture nel mezzo è 500 metri alla velocità di 10 Mbps, con un discreto rapporto di segnale/rumore, mentre per quanto riguarda la fibra ottica è necessario distinguere se si tratta di fibra in vetro o in plastica.
La lunghezza massima della connessione utilizzando la fibra multimodale in vetro è di 2000 metri a 100 Mbps full-duplex, mentre scende a poco piu di 100 metri utilizzando la fibra ottica in plastica nelle medesime condizioni.
La velocità di connessione offerta sulla stessa rete può quindi variare tra i valori 10 Mbps 100 Mbps e 1 Gbps.
Alcuni reti come SERCOS III non offrono questa possibilità e fissano un'unica velocità di connessione, scelta sicuramente dettata dal fatto che devono gestire un protocollo proprietario e non solamente il TCP/IP UDP/IP.
Altri come EtherNet/IP permettono addirittura all'utente della rete di compiere questa scelta, a seconda che desideri massimizzare la quantità di dati che è possibile spedire (Throughput) nel qual caso utilizzerà 1 Gbps. Oppure abbia necessità di garantire un livello di immunità ai disturbi elettrici migliore e una tolleranza migliore alla varianza delle specifiche del cavo utilizzato e quindi sceglierà velocità di comunicazione minori.
Oltre allo standard di comunicazione e alla velocità raggiungibile, in industria è particolarmente importante l'immunità ai disturbi. Sono quindi disponibili in commercio varie tipologie di cavo in rame per ogni standard di comunicazione a seconda della presenza e della modalità di posa dello schermo.
Un tipologia ampiamente consigliata dai produttori di reti è il cavo CAT5e, la sua struttura, schematizzato in figura A.1, è formata addirittura da due schermi, uno esterno che abbraccia tutti i conduttori, e altri quattro interni che avvolgono ciascuno due conduttori. In questo modo è possibile contrastare i disturbi provenienti dall'esterno e quelli che si creano tra i conduttori dello stesso.

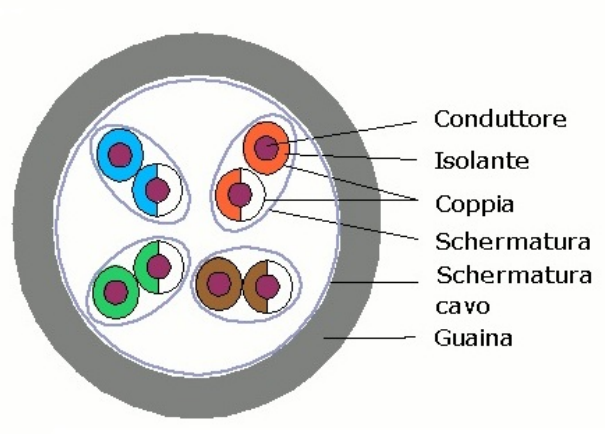


Fig. A.1: Cavo S/STP

Un'ultimo aspetto indispensabile per portare questo tipo di cavi in industria è la resistenza alle abrasioni, allo schiacciamento a al contatto con liquidi corrosivi.
In maniera simile ai cavi, anche i connettori presenti nei dispositivi si sono dovuti adattare ai nuovi standard di robustezza richiesti. Sono attualmente presenti in commercio connettori di tipo RJ45 o M12 specifici per ambienti industriali, che con meccanismi a ghiera o a incastro garantiscono la tenuta desiderata.

Un discorso a parte meritano gli standard Ethernet su fibra ottica, che grazie alla differente tecnologia trasmissiva non viene influenzata dal rumore elettromagnetico tipico del mondo industriale, per contro è molto piu delicata ai piegamenti e allo schiacciamento. Resta quindi una scelta valida in situazioni molto rumorose a patto di curare in maniera meticolosa la posa del cavo. I connettori specifici per fibra ottica sono i modelli SC, ST, MTRJ.

Appendice B

Programmi RSLogix 5000

Programma RSLogix 5000 per il calcolo della camma alpha

```
% Calcolo profilo di camma alpha
dimensione:= 150;
% numero di punti del profilo di camma

xc:=155;
yc:=-170;
raggio := 15;
% centro e raggio del cerchio

l1:=45.0;
l2:=230.0;
l3:=225.0;
l4:=45.0;
% lunghezza dei giunti del pentalatero

xa:=0;
ya:=0;
xb:=300;
yb:=0;
% coordinate dei due motori
% coincidenti con gli estremi vincolati
% del pentalatero

% calcolo del profilo
temp:=0;
for i:=0 to dimensione-1 by 1 do
t[i]:=temp;
x[i]:=(raggio * cos(t[i]))+ xc;
y[i]:=(raggio * sin(t[i]))+ yc;
temp:=temp +((2*3.14159265)/dimensione);

costeta2a := ((x[i]**2)+(y[i]**2)-(l1**2)-(l2**2))/
/(2*l1*l2);
senteta2a := - sqrt (1 - costeta2a**2);


if (l2*senteta2a)<0  then
cammaalpha[i].Slave:= ((- acos(x[i]/sqrt(x[i]**2+y[i]**2)) +
+(acos((l1+(l2*costeta2a))/
/sqrt((l2*senteta2a)**2+(l1+l2*costeta2a)**2)))))/(2*3.14159265);
end_if;
cammaalpha[i].Slave:= ((- acos(x[i]/sqrt(x[i]**2+y[i]**2)) -
+(acos((l1+(l2*costeta2a))/
```

```
/sqrt((l2*senteta2a)**2+(l1+l2*costeta2a)**2)))))/(2*3.14159265);

cammaalpha[i].Master:=t[i]/(2*3.14159265);
cammaalpha[i].SegmentType:=1;

end_for;
```

Programma RSLogix 5000 per il calcolo della camma beta

```
% Calcolo profilo di camma beta
dimensione:= 150;
% numero punti del profilo di camma

xc:=155;
yc:=-170;
raggio := 15;
% Centro e raggio del cerchio

l1:=45.0;
l2:=230.0;
l3:=225.0;
l4:=45.0;
% dimensione dei giunti

xa:=0;
ya:=0;
xb:=300;
yb:=0;
% coordinate dei due motori
% coincidenti con gli estremi vincolati
% del pentalatero

% calcolo del profilo di camma
temp:=0;
for i:=0 to dimensione-1 by 1 do
t[i]:=temp;
x[i]:=(raggio * cos(t[i]))+ xc;
y[i]:=(raggio * sin(t[i]))+ yc;
temp:=temp +((2*3.14159265)/dimensione);


costeta2b := (((x[i]-xb)**2)+((y[i]-yb)**2)-
+(l3**2)-(l4**2))/(2*l3*l4);
senteta2b := sqrt(1-costeta2b**2);


if (l3*senteta2b)<0  then
cammabeta[i].Slave:= ((- acos((x[i]-xb)/sqrt((x[i]-xb)**2+(y[i]-yb)**2)) -
+(acos((l4+(l3*costeta2b))
/sqrt((l3*senteta2b)**2+(l4+l3*costeta2b)**2)))))/(2*3.14159265);
end_if;
cammabeta[i].Slave:= ((- acos((x[i]-xb)/sqrt((x[i]-xb)**2+(y[i]-yb)**2)) +
```

```
+(acos((l4+(l3*costeta2b))/
/sqrt((l3*senteta2b)**2+(l4+l3*costeta2b)**2)))))/(2*3.14159265);


cammabeta[i].Master:=t[i]/(2*3.14159265);
cammabeta[i].SegmentType:=1;
end_for;
```