

## Libro bianco

# La sicurezza prima di tutto – determinazione della posizione e algoritmi di controllo di RESOLUTE™, encoder ottico assoluto-vero.

Questa informativa fornisce una panoramica del funzionamento dell'encoder assoluto RESOLUTE e descrive in dettaglio gli aspetti relativi alla sicurezza della determinazione della posizione e dei suoi algoritmi di controllo.

### Prefazione

La gamma degli encoder RESOLUTE, veri encoder assoluti, opera in modo radicalmente diverso rispetto ai tradizionali encoder assoluti, consentendo di emettere un flag di errore che indica con certezza se la posizione non è corretta. Ciò, fornisce un maggiore livello di sicurezza per gli utenti dei sistemi di movimento, semplificando nel contempo il processo di progettazione del costruttore del sistema.

RESOLUTE calcola la posizione su richiesta, a differenza degli encoder assoluti tradizionali che operano continuamente. Durante il funzionamento, il lettore riceve una serie di segnali di richiesta dal sistema di controllo host. Ogni volta che riceve una richiesta, il lettore determina la posizione con due metodi indipendenti (descritti in seguito) che si basano su principi completamente diversi, in modo da evitare un errore dovuto a cause comuni. Le posizioni ottenute vengono confrontate tra loro in modo che si possa decidere se generare un flag di errore da inserire nella posizione inviata al sistema di controllo. Ciò significa quindi che il flag di errore prodotto dal lettore risulta assolutamente affidabile per il sistema di controllo. Se non viene generato nessun flag, la posizione è sicuramente corretta.

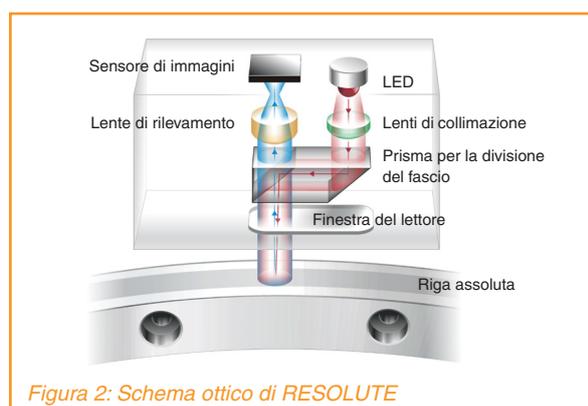


Figura 2: Schema ottico di RESOLUTE

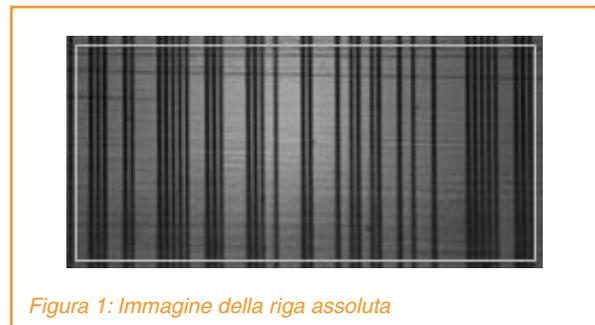


Figura 1: Immagine della riga assoluta

### Metodo per il calcolo della posizione

RESOLUTE è un encoder ottico che utilizza una riga di misura composta da linee scure su un substrato chiaro, brillante, come mostrato nella figura 1. Il periodo fondamentale della riga è di 30  $\mu\text{m}$ , ma alcune selezionate linee sono intenzionalmente mancanti al fine di codificare le informazioni della posizione assoluta.

Quando l'encoder riceve una richiesta di posizione, acquisisce un'accurata immagine istantanea di una zona della riga. Un LED ed un otturatore elettronico posto sul sensore dell'immagine (vedere Figura 2) vengono attivati per circa 100 ns. Il risultato è una nitida immagine acquisita con una temporizzazione compresa tra  $\pm 20$  ns. Tale immagine viene quindi trasferita ad un processore di segnale digitale (DSP) inserito nel lettore e la posizione viene calcolata utilizzando due algoritmi separati:

L'algoritmo 1 calcola la posizione decodificando una singola immagine, senza utilizzare alcuna informazione delle posizioni precedenti. La prima parte del processo consiste nel calcolare la fasatura dell'immagine. Questa operazione è simile a quella eseguita tramite encoder incrementali e produce una risposta pari ad una frazione del periodo fondamentale della riga ovvero un valore compreso fra 0 e 30  $\mu\text{m}$ , con una risoluzione migliore di 1 nm. Ogni immagine della riga si estende approssimativamente per 2 mm nella

direzione della misura e la riga è stata studiata in modo che ciascuna includa un numero di linee scure sufficiente per calcolare la fase con accuratezza. Il calcolo della fase consente di definire l'accuratezza sulle brevi distanze, la risoluzione e il rumore dell'encoder.

Le informazioni della fase vengono utilizzate per individuare il centro di ciascuna linea potenziale sulla riga. Viene quindi effettuata una correlazione sull'immagine per ogni posizione, allo scopo di verificare se sia presente o meno una linea scura. Il risultato è un numero binario a 65-bit che corrisponde allo schema della riga direttamente al di sotto del lettore. Solo un quarto di questi bits sono indispensabili per definire una posizione univoca. Il resto dei bits provvede a fornire informazioni ridondanti che permettono di accertare la corretta posizione anche se parte della riga risulta oscurata. Viene quindi utilizzato un algoritmo per il rilevamento e la correzione degli errori che consente di convertire la sequenza di bit nella posizione approssimativa assoluta del lettore. La posizione completa dell'algoritmo 1 si ottiene combinando la posizione approssimativa (numero totale di periodi della riga) con le informazioni sulla fase.

**L'algoritmo 2** calcola la posizione tramite estrapolazione lineare delle due letture più recenti della posizione. A tale scopo, si presume che la velocità dell'encoder rispetto alla lettura precedente sia la stessa di quella misurata fra le due posizioni precedenti. L'errore in questo calcolo viene determinato tramite il tempo fra le letture sequenziali, l'accuratezza delle letture precedenti, eventuali incertezze sui tempi e l'accelerazione relativa di lettore e riga. Per un sistema tipico che richiede un controllo della posizione ogni 62,5  $\mu\text{s}$ , con un'accelerazione massima fra lettore e riga di 100  $\text{m/s}^2$  (10 g), l'algoritmo 2 garantisce un errore massimo di  $\pm 1,2 \mu\text{m}$ . Per garantire che non vi siano mai più di 75  $\mu\text{s}$  fra le immagini, se necessario il lettore acquisisce immagini supplementari fra le richieste.

### Confronto fra le posizioni

Una volta che le due posizioni sono state calcolate, l'encoder prende una decisione relativamente alla posizione da produrre e sull'opportunità di generare un flag di errore. La parte della fase del calcolo dell'algoritmo 1 è molto solida. Se si introducono contaminazioni nella riga, le informazioni sulla fase diventano più rumorose e possono subire distorsioni. Tuttavia, anche in circostanze estreme, l'errore di posizione sarà inferiore a un micrometro. Le informazioni sulla fase risultano affidabili anche in presenza di contaminazioni significative, ma la capacità di correzione degli errori del calcolo della posizione approssimativa, potrebbe essere insufficiente per decodificare correttamente la posizione assoluta.

La posizione calcolata dall'algoritmo 2 fornisce una solida posizione approssimativa, perché solo un'accelerazione estrema ( $>2000 \text{ m/s}^2$ ) può causare un errore abbastanza significativo da modificare la posizione approssimativa. Tuttavia, la posizione precisa (fase) dell'algoritmo 2 non è particolarmente accurato perché presume che la velocità sia costante.

Se durante il funzionamento le posizioni approssimative calcolate con i due algoritmi combaciano, ovvero le posizioni finali si trovano entro  $\pm 15 \mu\text{m}$  (mezzo periodo di riga) l'una dall'altra, il lettore produce tale posizione approssimativa insieme alla fase ottenuta con l'algoritmo 1. Allo stesso tempo, un contatore interno viene impostato su zero (l'importanza del contatore risulterà chiara in un secondo momento). Se le posizioni approssimative non combaciano, il lettore produce la fase ottenuta con l'algoritmo 1 insieme alla posizione approssimativa dell'algoritmo 2 ed incrementa il contatore interno. Se il valore del contatore interno supera il valore di quattro, il lettore genera un flag di errore perché non esiste più la certezza che la posizione sia corretta. A questo punto, può essere utile prendere in esame alcuni esempi di cause che provocano la generazione di un flag di errore:

### Esempio 1

Presumiamo che il lettore passi su un'area in cui la contaminazione supera la capacità di correzione degli errori del codice della riga. L'algoritmo 1 produrrà una posizione approssimativa incorretta, ma la fase sarà esatta, anche se con un'inaccuratezza di qualche submicrometro dovuta alla presenza del contaminante. Il lettore registra internamente la discrepanza fra le posizioni approssimative (incrementando il contatore) e produce la posizione approssimativa corretta utilizzando l'algoritmo 2 e la fase esatta dell'algoritmo 1. Se il lettore non riesce a definire la posizione approssimativa corretta dall'algoritmo 1 per cinque immagini consecutive, imposta un flag di errore per indicare l'incertezza della posizione. Se l'algoritmo 1 riesce a definire correttamente la posizione approssimativa dopo non più di quattro immagini, il contatore viene azzerato e la procedura di determinazione della posizione continua.

### Esempio 2

Presumiamo che il lettore funzioni normalmente e quindi subisca un'accelerazione di  $\sim 10\,000 \text{ m/s}^2$ , equivalenti a una decelerazione da 2 m/s a fermo in 100  $\mu\text{m}$  (potrebbe essere causata da un arresto improvviso della macchina). In questo scenario la posizione dell'algoritmo 1 sarà sempre corretta, mentre quella dell'algoritmo 2 subirà un ritardo di pochi periodi di riga. Il lettore presume (sbagliando) che la posizione approssimativa dell'algoritmo 2 sia corretta e la

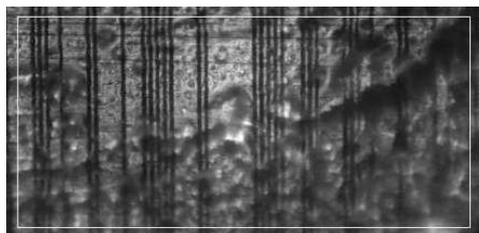
produce insieme alla fase corretta dell'algoritmo 1. Da questo momento in poi, le posizioni approssimative dei due algoritmi saranno sempre diverse. Prima di generare un flag di errore, il lettore calcola cinque posizioni che sono leggermente in ritardo rispetto alla posizione effettiva.

### Esempio 3

In questo esempio presumiamo che entrambi gli algoritmi siano in errore. Prendiamo una situazione altamente improbabile in cui radiazioni ionizzanti rovinano una parte della memoria all'interno del processore del lettore danneggiando le posizioni di entrambi gli algoritmi. Dato che gli algoritmi calcolano la posizione in modo completamente diverso, è impensabile che possano venire danneggiati in modo da produrre entrambi lo stesso identico errore. Pertanto, il lettore rileva una differenza nelle due posizioni approssimative e incrementa il contatore, producendo la posizione incorretta registrata dalla posizione approssimativa dell'algoritmo 2 e dalla fase dell'algoritmo 1. Poiché l'algoritmo 2 si basa sulle letture precedenti, da questo punto in poi tutte le risposte saranno errate. Per tale ragione, anche se l'algoritmo 1 riprende a fornire la risposta corretta, il lettore genera comunque un flag di errore dopo cinque posizioni incorrette.



*Figura 3: La contaminazione da particelle scure ampie regioni della riga, ma rimane possibile determinare informazioni sulla fase*



*Figura 4: La contaminazione da grasso può causare la diffusione complessa della luce, ma comporta solo un disturbo minimo alle informazioni sulla fase*

### Risposta al flag di errore

Gli esempi hanno dimostrato che il lettore può produrre fino a cinque posizioni incorrette prima di generare un flag di errore. Per un sistema che richiede una posizione ogni 62,5  $\mu\text{s}$ , il tempo fra la prima posizione incorretta e la segnalazione dell'errore è di 313  $\mu\text{s}$ . Nel caso di sistemi più lenti, che richiedono la posizione con intervalli di 500  $\mu\text{s}$  il tempo trascorso sarà di 500  $\mu\text{s}$ , perché il lettore deve elaborare sei immagini in più fra ciascun paio di richieste per accertare che il tempo trascorso fra due immagini non sia mai superiore a 75  $\mu\text{s}$ . In entrambi i casi, il tempo che intercorre fra la produzione della posizione non corretta e la generazione del flag di errore è sufficientemente breve da consentire di intraprendere le necessarie azioni correttive in risposta al flag di errore prima che i dati incorretti possano influenzare il sistema di controllo.

Le figure 3 e 4 mostrano alcuni esempi di contaminazione della riga che consentono comunque di determinare la fase, ma solitamente impediscono l'estrazione di un codice assoluto. In questi casi, i meccanismi descritti riusciranno a mantenere la posizione oppure porteranno alla generazione di un flag di errore.

### Accensione del sistema

Fino ad ora è stato dato per assodato che esistano sempre sufficienti dati storici per consentire all'algoritmo 2 di estrapolare la posizione. Questo caso non è applicabile alla prima accensione del lettore, perché non esiste una posizione estrapolata che possa essere confrontata con le letture della riga. In questo caso, vengono utilizzati due metodi per assicurare l'affidabilità della posizione del lettore in relazione alla riga.

Innanzitutto, il lettore imposta automaticamente il flag di errore se il contrasto dell'immagine risulta inaccettabile. In secondo luogo viene posta una limitazione sulla quantità di correzione errore consentita. Dato che il codice riga utilizza dati ridondanti per garantire che un ampio numero di bit risulti diverso fra le sequenze valide, tale limitazione sui bit correggibili serve sostanzialmente a limitare i rischi che una sequenza di bit possa essere erroneamente interpretata come una falsa posizione. Queste due limitazioni causano una leggera riduzione dell'immunità alla contaminazione del lettore al momento dell'accensione. Tuttavia, tale effetto viene controbilanciato dal grande numero di immagini acquisite dal lettore nella procedura di avvio che impediscono al rumore di limitarne la capacità di definire la posizione.

Inoltre, il codice riga è stato studiato in modo tale che, nel caso improbabile che il lettore determini una posizione

incorretta all'accensione, la discrepanza sarà rilevata entro 500 µm di spostamento. A quel punto verrà generato un flag di errore.

### CRC

Dopo che posizione e segnali di errore sono stati calcolati dal lettore, viene effettuato un controllo di ridondanza ciclico (CRC) da allegare ai dati prima della trasmissione al controllo host. La trasmissione avviene tramite segnalazione differenziale lungo un cavo a doppia schermatura. Dopo la ricezione dei dati, è possibile ricalcolare il CRC, confrontandolo con il valore trasmesso. Una differenza nei valori costituisce un'indicazione che i dati sono stati danneggiati durante la trasmissione. Questo meccanismo consente di rilevare eventuali danni ai dati sulla posizione o ai segnali di errore. Un ulteriore vantaggio derivante dall'utilizzo di protocolli seriali rispetto ai tradizionali sistemi di quadratura sta nel fatto che ciascuna trasmissione è indipendente, così gli errori di trasmissione non possono accumularsi. Questo vantaggio, unito all'esclusivo funzionamento di RESOLUTE,

fornisce agli utenti un enorme vantaggio per quanto riguarda la sicurezza, perché impedisce che si verifichino calcoli errati, spostamenti di posizione o conteggi fuori controllo.

### Conclusioni

Oltre a fornire eccellenti prestazioni metrologiche in condizioni normali, RESOLUTE ha un design pensato in modo specifico per garantire posizioni affidabili anche in presenza di errori o con un funzionamento anomalo. I designer di sistemi possono avere la certezza che RESOLUTE indica la posizione esatta o produce un flag di errore in caso di problemi. Gli utenti sono protetti da spostamenti fuori controllo e da rischi di collisioni, con vantaggi sulla resa, sulla produttività e cosa più importante sulla sicurezza.