

Artikel

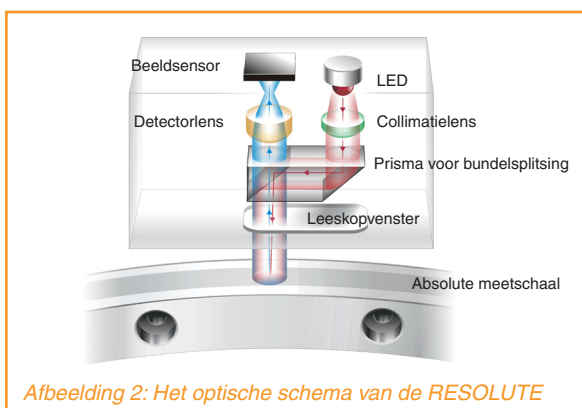
Veiligheid voorop - de algoritmes voor positiebepaling en-controle van de RESOLUTE™ werkelijk absolute optische encoder

Dit artikel vat samen hoe de RESOLUTE absolute encoder werkt en gaat gedetailleerd in op de veiligheidsaspecten rondom zijn algoritmes voor positiebepaling en -controle.

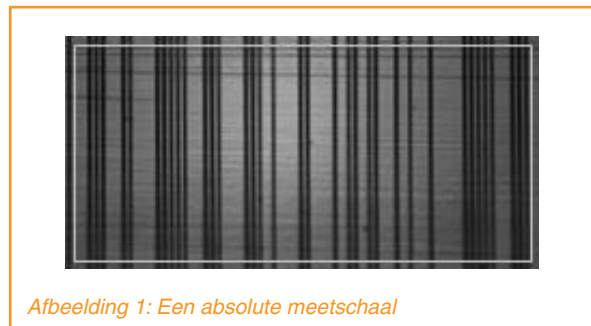
Inleiding

De RESOLUTE werkelijk absolute encoders werken op een fundamenteel andere manier dan traditionele absolute encoders, waardoor ze een foutmarkering kunnen geven en laten aanbrengen als de positiegegevens onjuist zijn. Dit verhoogt het veiligheidsniveau voor gebruikers van bewegingssystemen en vereenvoudigt het ontwerpproces voor de systeembouwer.

De RESOLUTE berekent de positie op verzoek, terwijl traditionele absolute encoders met een continue basis werken. Tijdens gebruik ontvangt de leeskop een reeks verzoeksignalen van het besturingssysteem. Bij ieder ontvangen verzoek bepaalt de leeskop de positie met twee onafhankelijke methodes, zoals verderop omschreven. Deze werken volgens geheel verschillende principes, wat het risico van dezelfde foutoorzaak voorkomt. De resulterende posities worden met elkaar vergeleken om te zien of er een foutmarkering moet komen bij de positie die naar het besturingssysteem gestuurd wordt. Het besturingssysteem kan dus vertrouwen op de foutmarkeringen die van de leeskop komen. Is er geen foutmarkering, dan staat vast dat de doorgestuurde positie juist is.



Afbeelding 2: Het optische schema van de RESOLUTE



Afbeelding 1: Een absolute meetschaal

Methode van positieberekening

De RESOLUTE is een optische encoder die een meetschaal gebruikt met donkere lijnen op een lichte ondergrond; zie afbeelding 1. De basis is een meetschaalperiode van 30 μm , maar een aantal lijnen is weggelaten om absolute positie-informatie te verkrijgen.

Wanneer de encoder een positieverzoek ontvangt, maakt hij een nauwkeurig getimede beeldopname van een gebied op de meetschaal. Een lichtgevende diode (LED) en een elektronische sluiters op de beeldsensor worden gedurende zo'n 100 ns in werking gezet; zie afbeelding 2. Dit resulteert in een beeld dat getimed is binnen ± 20 ns en vrij is van bewegingsonscherpte. Het beeld gaat vervolgens naar een digitale signaalprocessor (DSP) die zich in de leeskop bevindt, en de positie wordt berekend met twee verschillende algoritmes:

Algoritme 1 berekent de positie door één beeld te decoderen zonder gebruik te maken van eerdere positie-informatie. Als eerste stap van dit proces wordt de fase van het beeld berekend; deze stap lijkt op wat incrementele encoders doen. Het resultaat is een bepaalde fractie van de fundamentele meetschaalperiode, dus een waarde tussen 0 en 30 μm , met een resolutie die nog beter is dan 1 nm. Elk beeld van de meetschaal beslaat ongeveer 2 mm in de meetrichting, en de meetschaal is zodanig uitgevoerd dat elk beeld voldoende

donkere lijnen omvat om nauwkeurig de fase te berekenen. Deze faseberekening definieert de nauwkeurigheid over korte afstanden, de resolutie en het ruisniveau van de encoder.

Met de fase-informatie wordt het midden van iedere potentiële lijn op de schaal gelokaliseerd. Vervolgens wordt een correlatie met het beeld uitgevoerd voor elke van deze locaties, om vast te stellen of er een donkere lijn is of niet. Dit levert een 65-bits binair getal dat overeenkomt met het meetschaalpatroon direct onder de leeskop. Slechts een kwart van deze bits is nodig om een unieke positie te definiëren. De overige bits vormen dubbele informatie, zodat zelfs op een deels moeilijk zichtbare meetschaal de juiste positie te bepalen is. Daarna zet een foutdetectie- en correctiealgoritme de serie bits om naar een grove absolute positie van de leeskop. De uiteindelijke positie vanuit algoritme 1 volgt door de grove positie (een heel aantal meetschaalperiodes) te combineren met de fase-informatie.

Algoritme 2 berekent de positie door lineaire extrapolatie vanuit de twee recentste eerder afgelezen posities. Hiervoor wordt aangenomen dat de snelheid van de encoder na de laatste aflezing hetzelfde is gebleven als tussen de twee vorige posities werd gemeten. De fout in deze berekening wordt bepaald door de tijd tussen de opeenvolgende aflezingen, de nauwkeurigheid van de eerdere aflezingen, de spreiding in de timing en de relatieve versnelling van leeskop en meetschaal. Bij een gebruikelijk systeem dat om de 62,5 µs de positie opvraagt en een versnelling van leeskop ten opzichte van meetschaal heeft van maximaal 100 m/s² (10 g), levert algoritme 2 een fout van maximaal ±1,2 µm. De leeskop zorgt ervoor dat er nooit meer dan 75 µs tussen de beelden zit, door indien nodig tussen de positieverzoeken extra opnamen te maken.

Posities vergelijken

Zijn de twee posities eenmaal berekend, dan beslist de encoder welke positie hij rapporteert en of hij een foutmarkering geeft. Het fasegedeelte van de berekening met algoritme 1 is zeer robuust. Als de meetschaal vervuild raakt, krijgt de fase-informatie meer ruis en mogelijk ook vervormingen. Maar zelfs onder extreme omstandigheden blijft de hierdoor veroorzaakte positiefout beperkt tot minder dan een micrometer. De fase-informatie is nog steeds betrouwbaar wanneer er aanzienlijke verontreiniging is, maar de foutcorrectie bij het berekenen van de grove positie kan ontoereikend worden om tot de juiste absolute positie te komen.

Uit algoritme 2 volgt een robuuste grove positie, aangezien alleen extreme versnellingen (> 2000 m/s²) een dermate grote fout kunnen veroorzaken dat de grove positie anders wordt. Daarentegen is de fijne positie (de fase) van algoritme

2 niet al te nauwkeurig vanwege de aanname dat de snelheid constant blijft. (

Im Betrieb, wenn die Grobpositionen, die aus den zwei Als bij metingen de door beide algoritmes berekende grove posities overeenkomen, dan verschillen de uiteindelijke posities minder dan ±15 µm (de halve meetschaalperiode) met elkaar. In dat geval rapporteert de leeskop deze grove positie samen met de fase uit algoritme 1. Tegelijk wordt een ingebouwde teller op nul gezet; de rol van deze teller wordt verderop verduidelijkt. Komen de grove posities niet overeen, dan geeft de leeskop de fase van algoritme 1 samen met de grove positie van algoritme 2. Ook wordt de ingebouwde teller opgehoogd. Mocht de teller ooit de waarde vier overschrijden, dan geeft de leeskop een foutmarkering aangezien het niet langer zeker is dat de geleverde positie juist is. We gaan enkele voorbeelden bekijken van oorzaken die leiden tot een foutmarkering:

Voorbeeld 1

Stel dat de leeskop over een vervuild gebied beweegt dat de mogelijkheden van de foutcorrectie op de meetschaalcode te boven gaat. Dit resulteert in een onjuiste grove positie vanuit algoritme 1. De fase is echter correct, zij het met een sub-micrometer onnauwkeurigheid vanwege de aanwezige vervuiling. De leeskop registreert intern dat er verschil is tussen de grove posities (door zijn teller op te hogen) en geeft de juiste grove positie weer van algoritme 2 samen met de juiste fase van algoritme 1. Als het de leeskop bij vijf opeenvolgende beelden niet lukt om met algoritme 1 de juiste grove positie vast te stellen, dan volgt een foutmarkering om aan te geven dat de positie niet langer zeker is. Als algoritme 1 na vier of minder beelden de grove positie weer correct vaststelt, dan wordt de teller gereset en gaat de positieaanduiding door als tevoren.

Voorbeeld 2

Stel dat de leeskop normaal werkt en dan ineens een versnelling krijgt van zo'n 10.000 m/s². Dit komt overeen met afremmen van 2 m/s tot stilstand over 100 µm, wat zou kunnen gebeuren als een machine tegen een vaste eindaanslag loopt. In dit scenario is de positie vanuit algoritme 1 altijd correct, terwijl de positie vanuit algoritme 2 naloop op de juiste positie over hooguit enkele meetschaalperiodes. De leeskop neemt onterecht aan dat de grove positie van algoritme 2 juist is en rapporteert deze samen met de juiste fase van algoritme 1. Vanaf dit moment zullen de grove posities vanuit de beide algoritmes blijven verschillen. De leeskop berekent zo vijf posities die enigszins nalopen op de feitelijke positie en geeft dan een foutmarkering..

Voorbeeld 3

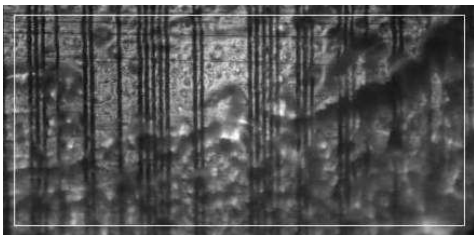
Stel in dit voorbeeld dat de twee algoritmes tegelijkertijd verkeerd rekenen. Dat kan bijvoorbeeld in het onwaarschijnlijke geval dat ioniserende straling het geheugen van de leesprocessor zodanig beschadigt dat de posities vanuit allebei de algoritmes niet meer kloppen. Omdat de algoritmes zo geheel verschillend hun posities berekenen, is het in feite onmogelijk dat ze zodanig beschadigd raken dat ze allebei hetzelfde verkeerde resultaat geven. De leeskop merkt dus op dat de grove posities verschillen, hoogt zijn teller op en rapporteert een onjuiste positie, samengesteld uit het grove deel van algoritme 2 en de fase van algoritme 1. Aangezien algoritme 2 uitgaat van eerdere aflezingen, zijn vanaf dit moment al zijn resultaten onjuist. De leeskop zal dus hoe dan ook een foutmarkering geven omdat hij vijf onjuiste posities registreert, zelfs als algoritme 1 zich herstelt en weer correcte resultaten gaat geven.

Reactie op foutmarkering

Zoals de voorbeelden omschrijven kan de leeskop hooguit vijf onjuiste posities geven voordat een foutmarkering ontstaat. Bij een systeem dat elke 62,5 μs de positie opvraagt, is de tijd tussen de eerste onjuiste positie en de foutmarkering 313 μs . Bij een langzamer systeem dat per 500 μs de positie opvraagt, bedraagt die tijd 500 μs , aangezien de leeskop



Afbeelding 3: Vuildeeltjes maken grote delen van de meetschaal moeilijk zichtbaar, maar fase-informatie is nog steeds te verkrijgen.



Afbeelding 4: Vervuiling met vet veroorzaakt een complexe lichtverstrooiing, maar de fase-informatie wordt maar minimaal verstoord.

tussen elke twee verzoeken al zes extra beelden verwerkt heeft om ervoor te zorgen dat de tijd tussen twee beelden nooit langer is dan 75 μs . In beide gevallen is de tijd tussen het geven van een onjuiste positie en het ontstaan van een foutmarkering voldoende kort om passend te kunnen reageren op de foutmarkering voordat de onjuiste gegevens het besturingssysteem gaan beïnvloeden.

De afbeeldingen 3 en 4 tonen voorbeelden van meetschaalvervuiling waarbij de fase nog te bepalen is maar absolute code niet meer haalbaar is. In zulke gevallen zullen de omschreven mechanismen ofwel de positie vasthouden ofwel daartegen waarschuwen door een foutmarkering te geven..

Systeem inschakelen

Tot nu toe hebben we steeds aangenomen dat er voor algoritme 2 altijd genoeg historische gegevens zijn om de positie te extrapoleren. Dit is echter meteen na het inschakelen van de leeskop niet het geval, want dan kan er nog geen geëxtrapoleerde positie zijn om de aflezing van de meetschaal mee te vergelijken. In deze situatie worden twee methodes gebruikt om de positie van de leeskop ten opzichte van de meetschaal betrouwbaarder te maken..

Ten eerste geeft de leeskop automatisch een foutmarkering als het beeldcontrast onacceptabel is. Ten tweede wordt aan de mate van foutcorrectie een maximum verbonden. De meetschaalcode gebruikt dubbele gegevens om zeker te stellen dat geldige reeksen veel van elkaar verschillen in bits. Een restrictie van de corrigeerbare bits beperkt daardoor het risico dat een bitreeks onjuist gedecodeerd wordt tot een verkeerde positie. Deze twee maatregelen maken de vuilbestendigheid van de leeskop iets lager tijdens het inschakelen. Dit effect wordt echter opgeheven doordat de leeskop tijdens zijn opstartprocedure een groot aantal beelden opneemt, zodat ruis geen beperking meer vormt bij het bepalen van de positie.

Bovendien is de meetschaalcode zo opgezet, dat in het zeer onwaarschijnlijke geval dat de leeskop bij inschakelen een onjuiste positie vaststelt het verschil wordt gedetecteerd binnen een verplaatsing van 500 μm . Op dat moment volgt een foutmarkering.

CRC

Heeft de leeskop eenmaal de positie en foutsignalen vastgesteld, dan wordt een cyclische dubbelcontrole (CRC) uitgevoerd en aan de gegevens gekoppeld voordat deze naar de besturing gaan. De overdracht gebeurt met differentiële signalen via een dubbel afgeschermd kabel. Na aankomst van de gegevens wordt de CRC opnieuw berekend en

vergeleken met de overgedragen waarden. Een verschil tussen deze waarden geeft aan dat de gegevens tijdens de overdracht beschadigd zijn. Zo is het zeker dat elke beschadiging van de positie- of foutsignalen gedetecteerd wordt. Nog een voordeel van seriële protocollen ten opzichte van traditionele kwadratuursystemen is dat elke overdracht onafhankelijk is, waardoor fouten in de overdracht zich niet kunnen opstapelen. Samen met de unieke werking van de RESOLUTE ontstaat zo een essentieel veiligheidsvoordeel, want telfouten, positieverloop en een 'weglopende' telling kunnen niet plaatsvinden.

Conclusies

Naast zijn uitstekende meetprestaties onder normale omstandigheden is de RESOLUTE specifiek geoptimaliseerd om ook in moeilijke of afwijkende situaties betrouwbare positiegegevens te garanderen. Systeemontwerpers kunnen erop vertrouwen dat de RESOLUTE de werkelijke positie rapporteert of anders het tegendeel kenbaar maakt. Gebruikers zijn beveiligd tegen ongecontroleerde verplaatsingen en botsrisico's en verkrijgen meer rendement, capaciteit en vooral veiligheid.