Mikroprocessorn i servo-tekniken

Sammanfattning

Dagens ökande krav på prestanda och flexibilitet hos servosystem, kombinerat med den ökade kapaciteten hos dagens mikroprocessorer leder till att allt mer av den grundläggande motorstyrningen i ett servosystem utförs i mjukvara i stället för hårdvara.

Från det traditionella systemet, där mikroprocessorn huvudsakligen står för positions-loopen går utvecklingen mot att den även får ta hand om hastighets-loop, eventuell fas-fördelning och mätvärdesovandling.

Vi kommer att titta på ett modernt, resolverbaserat servosystem avsett för i första hand PM-synkronmotorer, samt göra en del jämförelser med den traditionella regulatorn.

Det system vi tittar på är ett enaxligt system, såtillvida att varje processor bara kan styra en axel. Däremot kan flera system kopplas samman, så att en fleraxlig styrning erhålles.

Lämpliga mikroprocessorer

Normalt får man räkna med att man minst behöver en 16 bitars processor, om man inte avser att bygga ett traditionellt system. Det är viktigt att processorn har MUL-instruktion, och gärna även DIV.

Vill man styra flera axlar med en processor eller har väldigt höga prestandakrav använder man vanligen en digital signalprocessor (DSP).

Vill man ha ett billigt och kompakt servosystem med inte alltför höga prestanda väljer man vanligen en 'controller-krets'.

Systemets delar

Ett modernt servosystem består av (se bild 1):

- ☐ Profilgenerator (börvärdesgenerator/interpolator)
- ☐ Positions och hastighetsregulator
- ☐ Kommutering/fasfördelningsenhet
- ☐ Strömregulator
- ☐ Effektsteg för drivning av motorn
- ☐ Anpassning till positions- och hastighetsgivare
- ☐ In- och utgångar (I/O) för givare och enklare kommandon
- ☐ Ett programmeringsspråk för att styra applikationen
- ☐ Datainsamlingsrutiner för felsökning och trimning
- ☐ Eventuellt nätverk för höghastighetskommunikation med andra system

Till detta behövs också:

- ☐ Programutvecklingsmiljö för utveckling och dokumentation av styrprogram
- ☐ Program för analys av insamlade data

Profilgeneratorn

Profilgeneratorns uppgift är att generera den "bana" som rörelsen skall följa när man vill nå en viss punkt.

För enaxliga servosystem inskränker sig detta normalt till att gå till en viss punkt, men med en viss maximal hastighet och en viss maximal acceleration/retardation. Det är också vanligt att acclerationens derivata ("jerk" eller "ryck") är begränsad, en så kallad s-profil.

För fleraxliga system är profilgeneratorn avsevärt mer komplex än för enaxliga. Den kan t ex behöva dra cirklar i rymden, samtidigt som den iakttager alla ovanstående begränsningar. Om en motor i ett system (t ex en robot) skulle behöva överskrida något maximalt tillåtet värde, måste alla motorer saktas ned så att rörelsen fortfarande blir en cirkel.

Det är också vanligt att en fleraxlig profilgenerator klarar omvandlingar mellan olika koordinatsystem, t ex till och från polära koordinater.

Om en viss rörelse alltid ser ungefär likadan ut kan den beräknas utanför servosystemet och läggas upp i en tabell som ett antal värden som sedan profilgeratorn bara har att interpolera mellan. Index-positionen i tabellen kan antingen vara tidsstyrd eller tas från någon ingång hos systemet, t ex en analogingång, eller från en extra positionsgivare.

Om man vill köra fleraxligt med system med profilgeneratorer som inte klarar flera axlar kan man förbikoppla den inbyggda profilgeneratorn och ta börvärden direkt utifrån, från en yttre profilgenerator, t ex via nätverket.

Regulatorn

Nu skall alltså motorns hastighet eller position följa de värden som kommer från profilgeneratorn så bra som möjligt. Detta innebär att motorns avgivna mekaniska moment hela tiden måste beräknas, med målet att hålla motorns position och eller hastighet så nära den önskade som möjligt.

Detta är regulatorns uppgift. Den har en insignal, som kan vara önskad position (X) eller önskad hastighet (V) och ger en utsignal; det önskade momentet.

Detta är bara en av flera delar i den loop som består av motor, eventuell växel, den drivna maskinen, samt någon form av givare och dess elektriska anpassning. Prestanda hos det reglerade systemet beror lika mycket på de övriga elektromekaniska komponenterna som på själva regulatorn.

Ingen regulator kan garantera total avsaknad av följ-fel, men de går att få ganska små. Felen beror på:

- ☐ Konstruktionsprincip
- ☐ Hur pass välgjord regulatorn är
- ☐ Kvaliteten på den återkopplade positionen och hastigheten
- ☐ Regulatorns justering

Varför måste man justera?

Avsikten med regulatorn är att garantera en viss noggranhet.

Trots att prestanda hos hela det reglerade systemet beror på varje element i hela loopen, är regulatorn normalt det enda elementet som kan påverkas. Detta innebär att dess egenskaper måste tillmötesgå två krav. Den måste hålla följ-felet till ett minimum och samtidigt inte åstakomma några instabiliteter.

Beteendet hos det reglerade systemet och nogrannhetskraven kan variera väldigt mycket från en applikation till en annan.

I reglerteorin finns diverse metoder för att med hjälp av en modell beräkna regulatorns egenskaper, baserat på data om systemet som skall regleras. Resultatet från dessa modeller får ofta tas med en nypa salt, eftersom de har en tendens till att vara alltför optimistiska. I modellerna ingår ofta inte glapp i växlar, fjädringar, resonansfrekvenser, olinjäriteter och andra fenomen som förekommer i verkligheten.

Därför måste varje regulator slutjusteras mot varje ny applikation, genom ett test i full skala.

Traditionell regulator

Traditionellt utförs positionsreglering av en motor med hjälp av tre hopkopplade reglerloopar enligt bild 2.

Den första loopen reglerar motorströmmen Im på ett sådant sätt att den uppmätta strömmen är nästan samma som den önskade Ib, oberoende av påverkan från:

- ☐ Variationer i inducerad motorspänning (hastighetsberoende)
- ☐ Variationer i matningsspänningen
- ☐ Impedansen i motor och kablar

En andra loop reglerar motorns hastighet. Dess uppmätta hastighet, Vm (fås normalt från en tacho-generator), jämförs med den önskade Vb. Skillanden dV behandlas av regulatorn och genererar den önskade strömmen Ib, till strömregulatorn.

Meningen med denna regulator är att hålla nere hastighetsavvikelserna så mycket som möjligt, även under påverkan av:

- ☐ Tröghetsmomentet vid hastighetsändringar
- ☐ Friktion
- ☐ Laständringar
- ☐ Variationer i motorkarakteristik
- ☐ Ofullständigheter i strömregulatorn

En tredje loop reglerar positionen på det drivna systemet. Positionen Xm mäts med en inkrementell pulsgivare monterad på motoraxeln.

I motsats till de analoga ström- och hastighetsregulatorerna är positionsregulatorn till sin natur digital. Positionen mäts inte kontinuerligt utan med ett noggrant bestämt intervall. Vid varje sådant intervall, eller "servocykel" jämförs den uppmätta positioen Xm med den önskade Xb, som i vårt fall kommer från en profilgenerator. Avvikelsen dX behandlas av positionsregulatorn för att ge den önskade hastigheten Vb till hastighetsregulatorn.

Målet är vanligtvis att hålla positionsavvikelsen så låg som möjligt även under yttre påverkan. Den påverkan som normalt kommer fram till positionsregulaton är i huvudsak orsakad av ofullständigheter i hastihgetsregulatorn.

Det största problemet med att förbättra prestanda hos dessa traditionella regulatorer är det faktum att de är förpackade i två separata enheter. Ett positioneringssystem eller en CNC-styrning hanterar positionsregleringen, medan en servo-drive hanterar ström- och hastighetsregleringen. Sammankopplingen sker med en +–10 V analog signal, vilken överför den önskade hastigheten Vb.

Om man betraktar vilken påverkan de olika regulatorerna skall kunna kompensera för, ser man att strömregulatorn skall kompensera för förstärkar- och motorberoende effekter. Är den en gång korrekt inställd för en kombination av motor/förstärkare/nätaggregat, fungerar inställningen med all sannorlikhet i alla applikationer där denna konfiuguration kan användas. Hastighetsregulatorn däremot, skall kunna kompensera för de applikationsberoende effekterna, som tröghetsmoment, friktion, last etc. Positionsregulatorn till sist, skall korrigera för ofullständigheterna i hastighetsregulatorn. Detta gör att man normalt inte tjänar något på att använda en mer avancerad positionsregulator än en vanlig PI. Vanligt förekommande är också enbart P eller en P-regulator med lead-lag-filter.

Moderna regulatorer

För att kunna förbättra prestanda hos positionsregulatorn krävs att positions- och hastighetsregulatorerna integreras i en och samma enhet. (Strömregulatorns inställning påverkas inte nämvärt av applikationen, så den kan lämnas därhän tills vidare.) Denna integrering av position och hastighetsregulatorerna kräver avsevärt högre prestanda hos den mikroporcessor som styr systemet. Dock öppnar detta vägen för de mest avancerade regleralgoritmerna, för automatisk identifikation av systemets karakteristik samt för adaptiva regleralgoritmer.

Framkopplad regulator (Feed Forward)

Integreringen av positions- och hastighetsregulatorerna möjligör också användandet av framkoppling. Om man har ett system med relativt välkända egenskaper kan man kompensera för dessa genom så kallad framkoppling (Feed Forward). Se bild 3.

Den övre halvan med faktorerna K-Pos, K-integr och K-hast motsvarar P I D i en vanlig regulator. Om man känner till tröghetsmoment, torr och viskös friktion samt yttre extra moment på systemet, är det möjligt att beräkna det erfoderliga momentet hos motorn. Detta resulterar i att PID-delen av regulatorn bara behöver kompensera för skillnaden mellan de uppskattade värdena och de verkliga.

Det finns också ett flertal begränsningar inlagda i regulatorn. Begränsningen MAX Ström/moment kan inte bara användas till att skydda motorn, utan också om man vill kunna köra momentstyrt istället för positions/hastighetsstyrt.

De tre övriga begränsningarna kan bl a användas för att begränsa det av regulatorn genererade momentet, utan att begränsa det framkopplade momentet. Om man i en applikation t ex behöver 10 Nm för en acceleration, men motorn maximalt klarar 2 Nm kontinuerligt, kan man begränsa regulatorn till att ge maximalt 2 Nm, men att under själva rörelsen ge motorn 10 Nm via framkopplingen.

Då alla dessa konstanter inte är trimpotentiometrar eller kondensatorer på förstärkaren, utan variabler som är åtkomliga för applikationsprogrammet, även under drift, öppnar det möjligheten för adaptiva regulatorer och automatisk trimning.

Fasfördelning/kommutering

Detta block behövs endast för AC-motorer, som PM-synkronmotorer och asynkronmotorer. För DC-motorer kan man gå direkt till slutsteget. Se bild 4.

Här görs den skalära storheten "Önskad ström" om till en strömvektor. Det sker med hjälp av vinkelinformation som hämtas från en kommuteringsgivare, i detta fall en resolver. I princip går det till så att vinkelvärdet från kommuteringsgivaren används för att slå upp värden ur två sinustabeller. Dessa värden multipliceras med den beordrade strömmen B-I och resulterar i börvärdesströmmarna B-Ir och B-Is till slutsteget. Slutsteget skapar själv t-fasen ur de två andra, varför bara två faser behöver genereras.

För att justera kommuteringsvinkeln finns två variabler. Den ena är avsedd för statisk korrigering. Att ändra på den motsvarar att vrida på resolvern. Den andra variablen används för hastighetsberoende justering av kommuteringsvinkeln. Detta för att förbättra prestanda vid höga varvtal, framförallt för att kompensera för fördröjningen i strömregulatorerna i slutsteget.

Blocket "Prediktering av rotorläge" används eftersom fasfördelningsrutinen körs fyra gånger för varje gång som ett nytt resolverläge beräknas. Den kompenserar också för fördröjningen mellan resolveravläsnigen och själva kommuteringen.

Slutligen Digital/Analogomvandlas signalerna och matas ut till det strömstyrda slutsteget.

Strömstyrt slutsteg

I slutsteget finns den innersta regulatorn i systemet, nämligen själva strömregulatorn. Se bild 5.

Hit har inte mikroprocessorn kommit ännu. Strömregulatorn är idag oftast en analog konstruktion. Det beror på flera saker: bland annat är detta den snabbaste reglerloopen i hela systemet. Den skulle därför ta en oproportionerligt stor del av processortiden i anspråk. Den tiden läggs med fördel på övriga funktioner i systemet. Dessutom är de snabba analog/digitalomvandlare som krävs för mätningen av strömmen relativt dyra. Till sist är den väldigt enkel att utföra i hårdvara.

De slutsteg som i dag styrs direkt av en mikroprocessor är i huvudsak av karaktären spänningsstyrda, som till exempel frekvensomriktare.

Två faser regleras efter inkommande börvärden och den tredje genereras som negativa summan av de övriga. De resulterande signalerna går till varsin PWM-generator. Signalen jämförs där med en triangelvåg, normalt på mellan 5 och 20 Khz. Resultatet blir en kantvåg med varierande pulsbredd. Denna signal matas sedan till respektive slutsteg.

Den förstärkta kantvågen matas vidare till motorn. Den är nu en kantvåg med ofta över 300 Volts spänning, mde stig- och falltider neråt 100 ns, med triangelvågens frekvens. Se upp med störningar! Det kan tyckas bekvämt att dra alla signaler till motorn i samma kabel, men då får man ta i beaktande att en differentiell störning på resolversignalerna om bara några 10-tal millivolt är tillräckligt för att försämra prestanda. Ju högre resolverupplösning, desto störkänsligare blir systemet.

Resolver-digital-omvandling

Detta block har till uppgift att mäta rotorläge och hastighet hos motorn. Se bild 6.

Resolvern kan förenklat ses som en vridtransformator med en primärlindning och två sekundärlindningar. Antag att man matar primärlindningen med en sinus-spänning och sedan tittar på utspänningarna vid olika vridningsvinklar (alfa) på axeln. För en tvåpolig resolver blir då den ena utspänningens amplitud proportionell mot sinus(alfa) och den andra mot cosinus(alfa).

I vårt fall genererar mikroprocessorn en synksignal (kantvåg) som matar en sinusgenerator, som i sin tur matar resolvern.

De resulterande singalerna filtreras och buffras för att sedan matas in i en hållkrets. En signal (S/H) från mikroprocessorn bestämmer när värdena från resolvern ska låsas. De ska låsas när de har sina max- respektive min-värden. De värden som man då har fått, motsvarar amplituden för respektive lindning. Tidpunkten för låsningen relativt den genererade synksignalen beror på fasförskjutning i resolver, kablage och ingångsfilter och måste normalt justeras individuellt för varje applikation, vilket kan göras automatiskt av mikroprocessorn vid systemstart.

De låsta signalerna analog/digitalomvandlas, varefter vinkeln alfa beräknas. Signalerna divideras först med varandra för att bli oberoende av dämpningen i resolvern och i kablaget, varefter man beräknar arctangens på resultatet och erhåller vinkeln alfa.

Denna vinkel är absolut inom ett elektriskt varv ifall motor och resolver har samma poltal och kan direkt användas för kommutering av motorn.

Prestanda

Med 8 bitars A/D-omvandlare når man en upplösning på ca 500 steg per elektriskt varv, och med 16 bitars, en upplösning på ca 120.000 steg/varv. Noggrannheten hos de vanligaste resolvrarna ligger runt +/–3.5 till +/–15 bågminuter, vilket motsvarar ca 720 till 3.000 steg/varv eller 9 till 11 bitars A/D.

Om man antar att de resulterande signalerna från resolvern ligger runt 10 Volt, kan man räkna ut de störnivåer som är på gränsen till att märkas. För 8 bitar blir det ca 80 mV, för 12 bitar ca 5 mV och för 16 bitar ca 0.3 mV.

I jämförelse med en 5-V differentiell pulsgivare där det behövs 2.5 volt för att störa ut signalen, kan dessa nivåer tyckas alarmerande. Man ska då komma ihåg att resolvern är absolut inom ett varv, och alltså kommer att återgå till att visa den riktiga positionen när störningen försvinner, såvida störningen inte är så kraftig att man tappar ett helt varv. För detta behövs en störning på i storleksordningen 5 Volt.

Om man antar en 10 bitars A/D-omvandlare och 30.000 rpm, motsvarar det ca 1 MHz från en pulsgivare (efter multiplikation). Bandbredden hos pulsgivaringången bör vara minst 2 MHz för att säkert kunna ta hand om denna frekvens. Resolveringången däremot, kan i detta fall klara sig med så låg bandbredd som 2 KHz.

Slutsaten blir att man för pulsgivaren tappar position utan egentligen märka några symptom, medan man i resolverfallet märker "darrig" gång hos systemet långt innan man riskerar att tappa positionen.

Användardefinerat applikationsprogram och datainsamling

För att systemet slutligen skall vara så använbart och flexibelt som möjligt, gäller det att det finns ett kraftfullt och lättförståeligt applikationsspråk som gör det enkelt för användaren att förstå och använda systemet. Det som gäller i dag är BASIC-liknande språk för programmering direkt i målsystemet, och mer avancerade grafiska programmeringsmiljöer för programutveckling på en värddator.

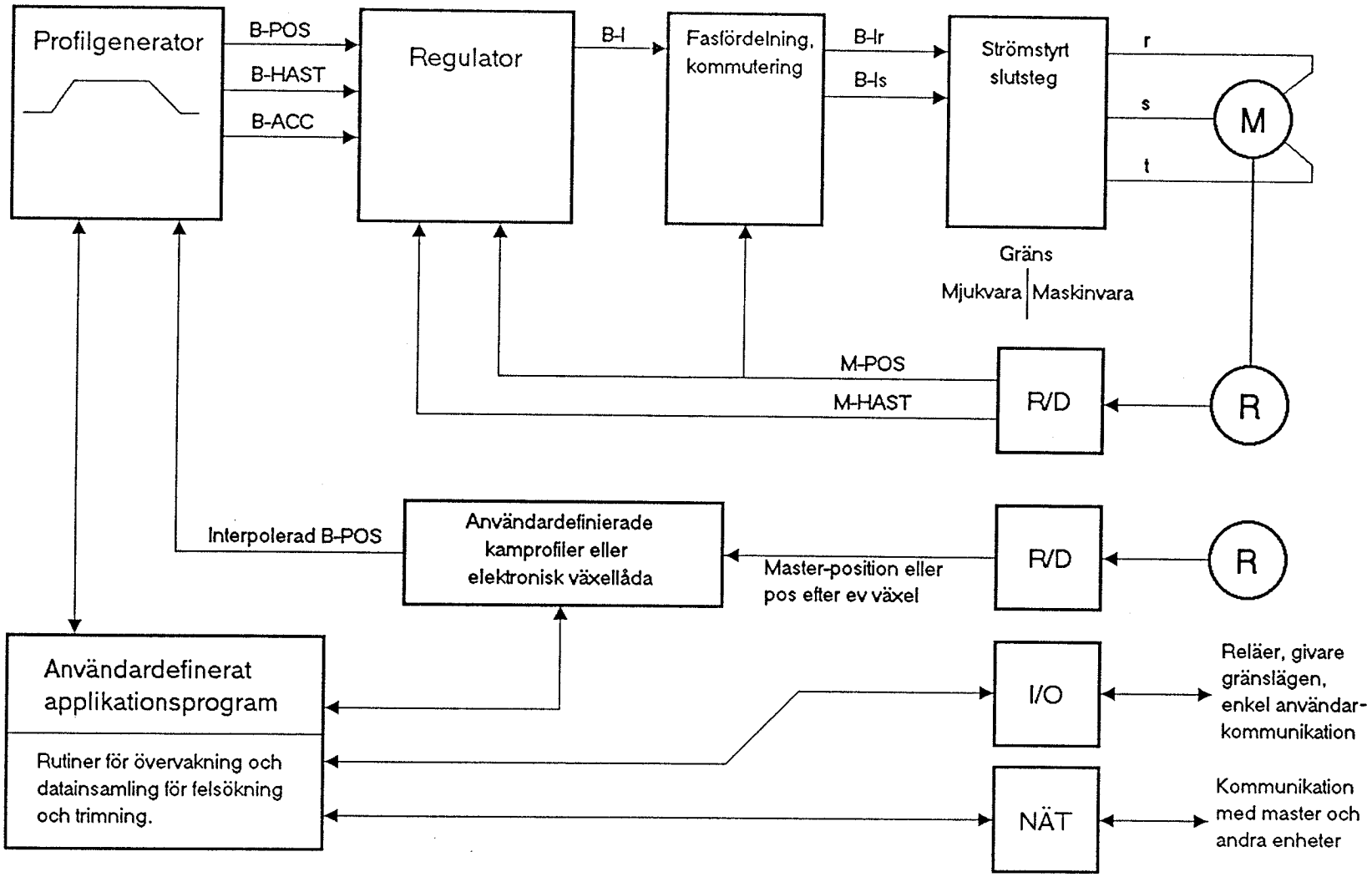
Fördelen med de BASIC-liknande språken är att de är lätta att göra interaktiva, så att man kan prova och ändra sina rutiner direkt på systemet utan att behöva kompilera om dem varje gång.

Vad gäller datainsamling blir det numera allt vanligare att man kan mäta signaler i systemet i realtid och lagra undan i en buffert för senare analys. I och med att systemen har blivit allt mera integrerade kan man inte längre komma åt intressanta mätpunkter i regulatorerna utifrån, om man inte bygger in den möjligheten från början. Det gör att dessa rutiner ofta byggs in av tillverkaren för att underlätta utveckling, prov och justering av olika regulatorer.

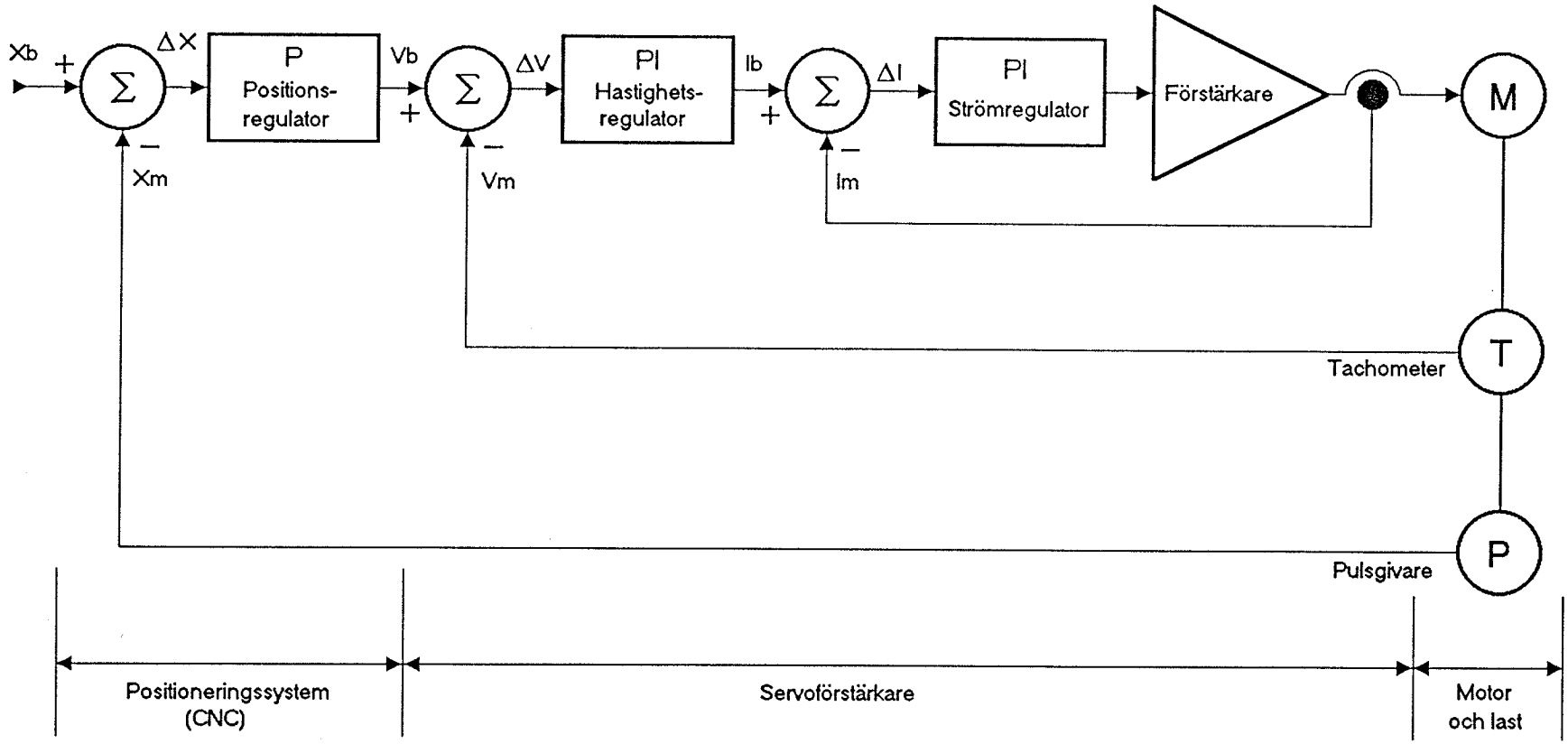
Vid justering av regulatorn kan man t ex lagra hastighet och position under en rörelse för att sedan analysera responsen i en värddator, eller måhända bara använda resultatet till dokumentation.

Om man låter datainsamlingen styras av vanliga programsatser så kan även maskinbyggaren uttnyttja dem för utveckling, service och underhåll av slutprodukten.

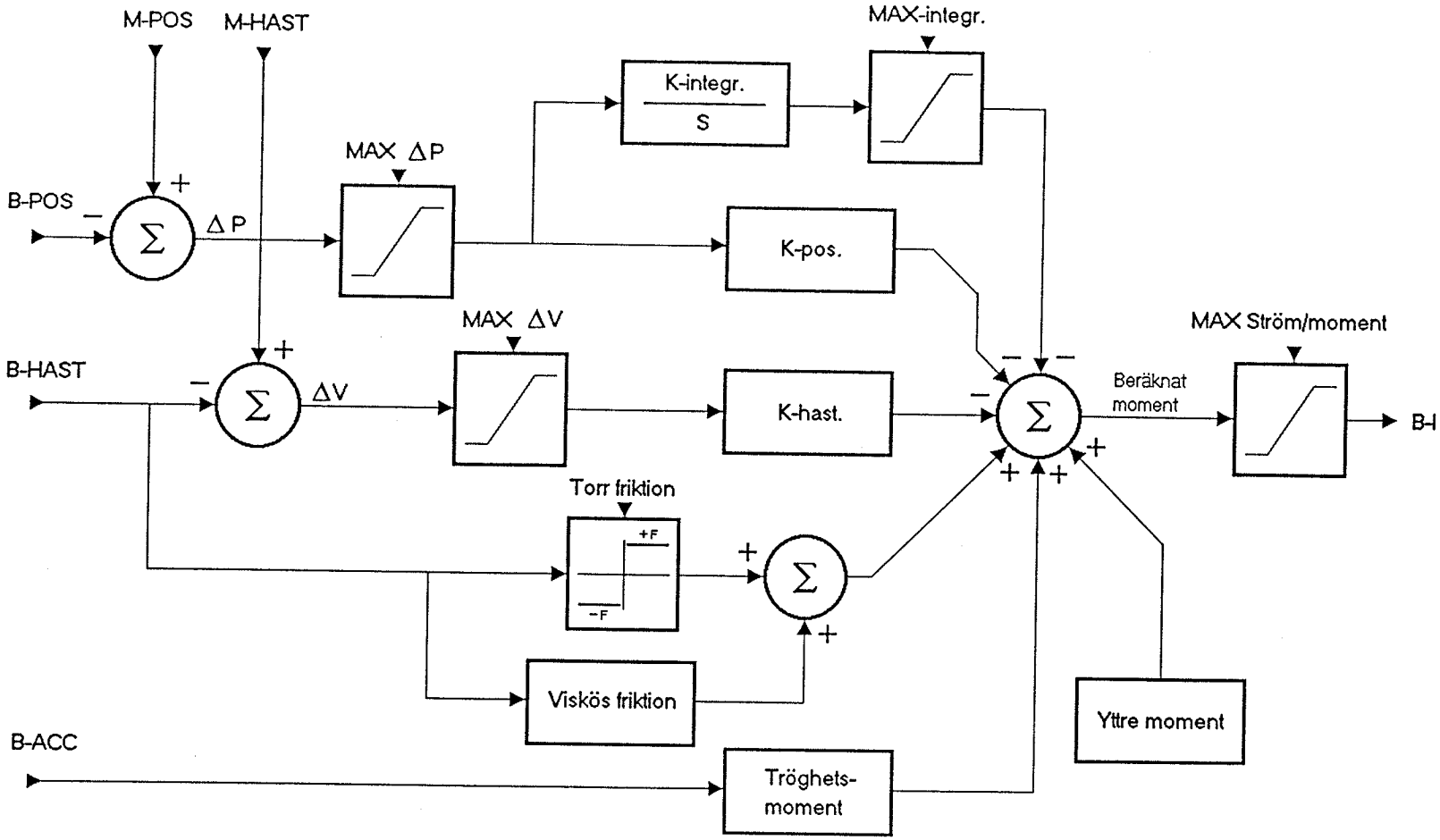
Blockschema, resolverbaserat servo



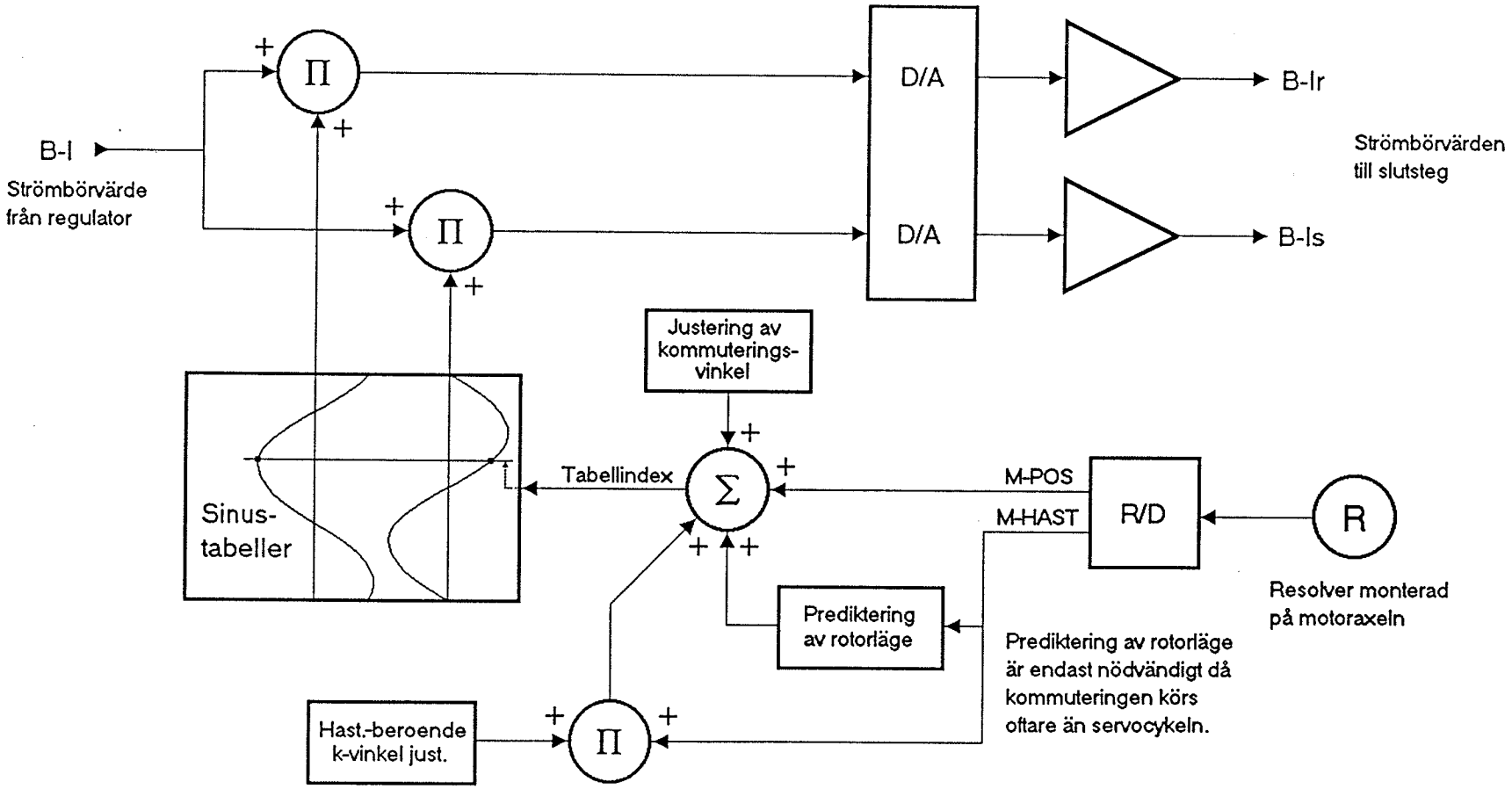
Traditionell positionsregulator



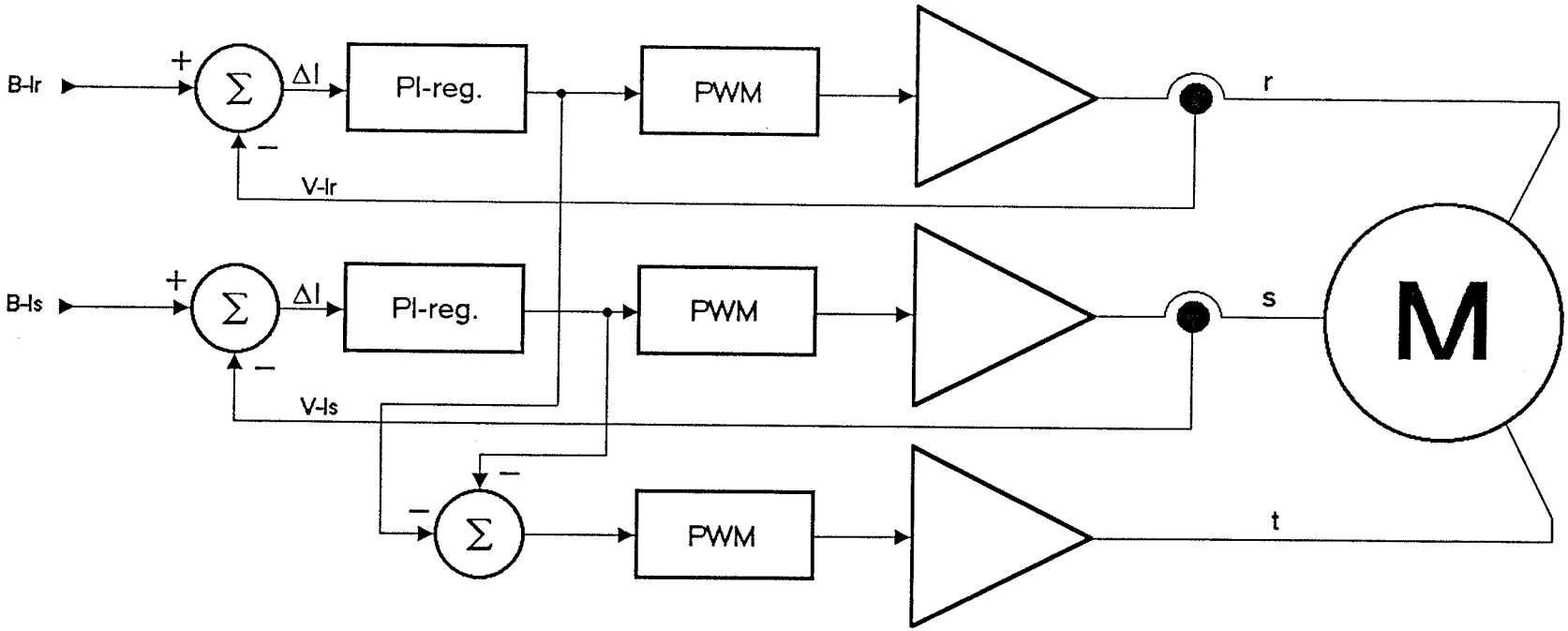
Regulator med framkoppling



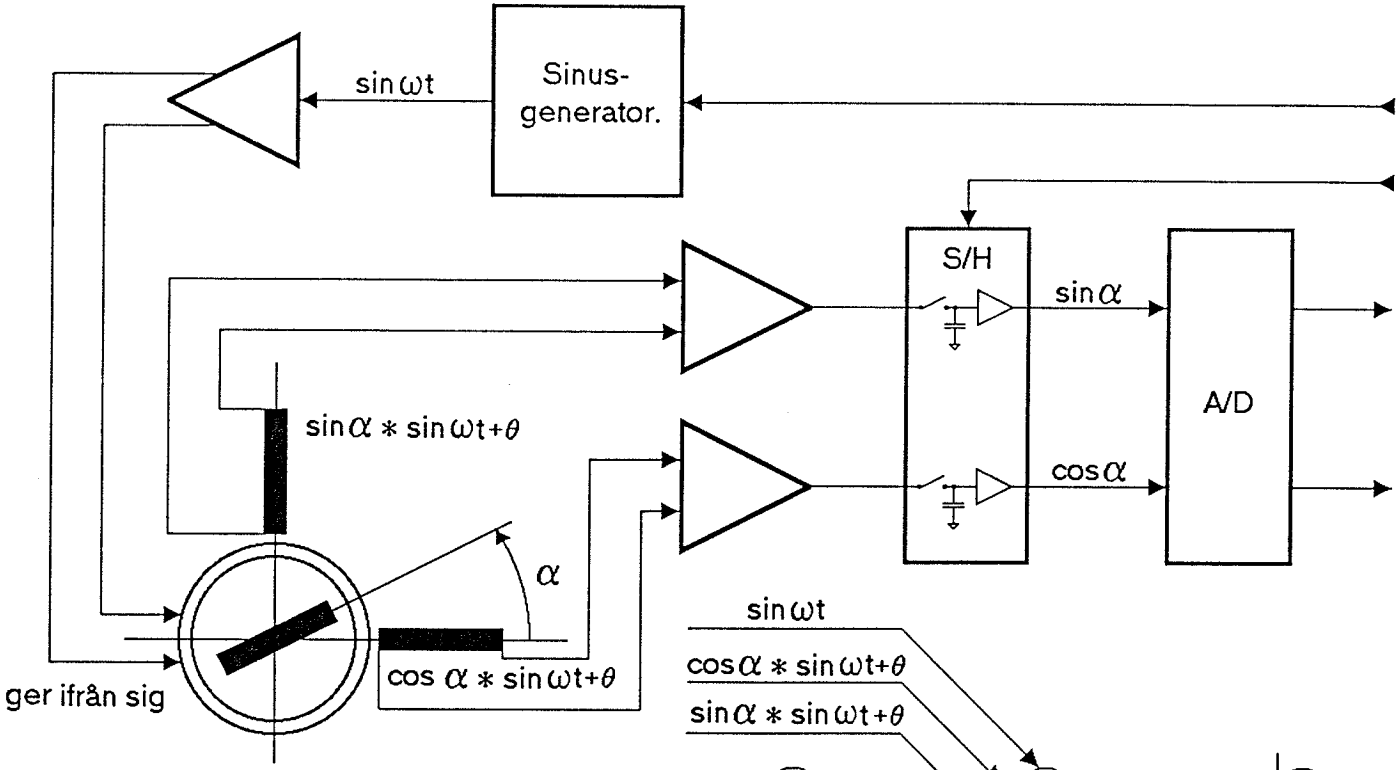
Fasfördelning/kommutering för synkronmotor



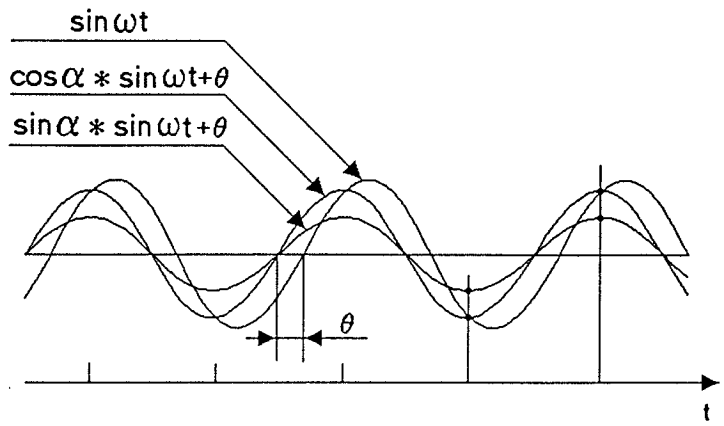
Strömstyrt slutsteg



Resolver / digitalomvandling med mikroprocessor



Resolvern exiteras med signalen sin ωt, och matar ut singnalerna ($\sin\alpha * \sin\omega t+\theta$) respektive ($\cos\alpha * \sin\omega t+\theta$).
För en tvåpolig resolver är vinkeln α lika med den sökta mekaniska vinkeln.



Signaler till och från mikro-processorn:

Referenssignal för resolver-exitering

Sample/Hold signal, korrigerad för θ. θ beror huvudsakligen på resolvertyp och kablage

Ur signalerna sinα och cosα beräknas tanα enligt:

$$\tan\alpha = \frac{\sin\alpha}{\cos\alpha}$$

Därefter kan man beräkna:

α = arctan(tanα) → M-POS
$V = \alpha_n - \alpha_{n-1}$ → M-HAST

Dessa beräkningar måste utföras minst en gång per servo-cykel och resolver, vilket kräver relativt mycket processorkraft.

Vid 1 KHz exiteringsfrekvens, 10 bitars A/D och 30.000 rpm motsvarar denna omvandling en pulsgivarfrekvens om ca 1 MHz.

Möjliga samplingspunkter