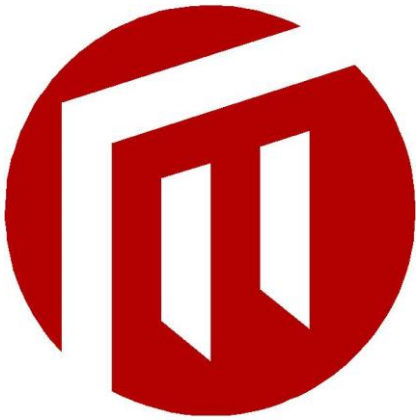


MALMÖ HÖGSKOLA

Inbyggda system och signaler

Styr- och reglerteknik

Examinationsuppgift

Namn: Stefan Angelov

Gion Koch Svedberg

26 Augusti 2015

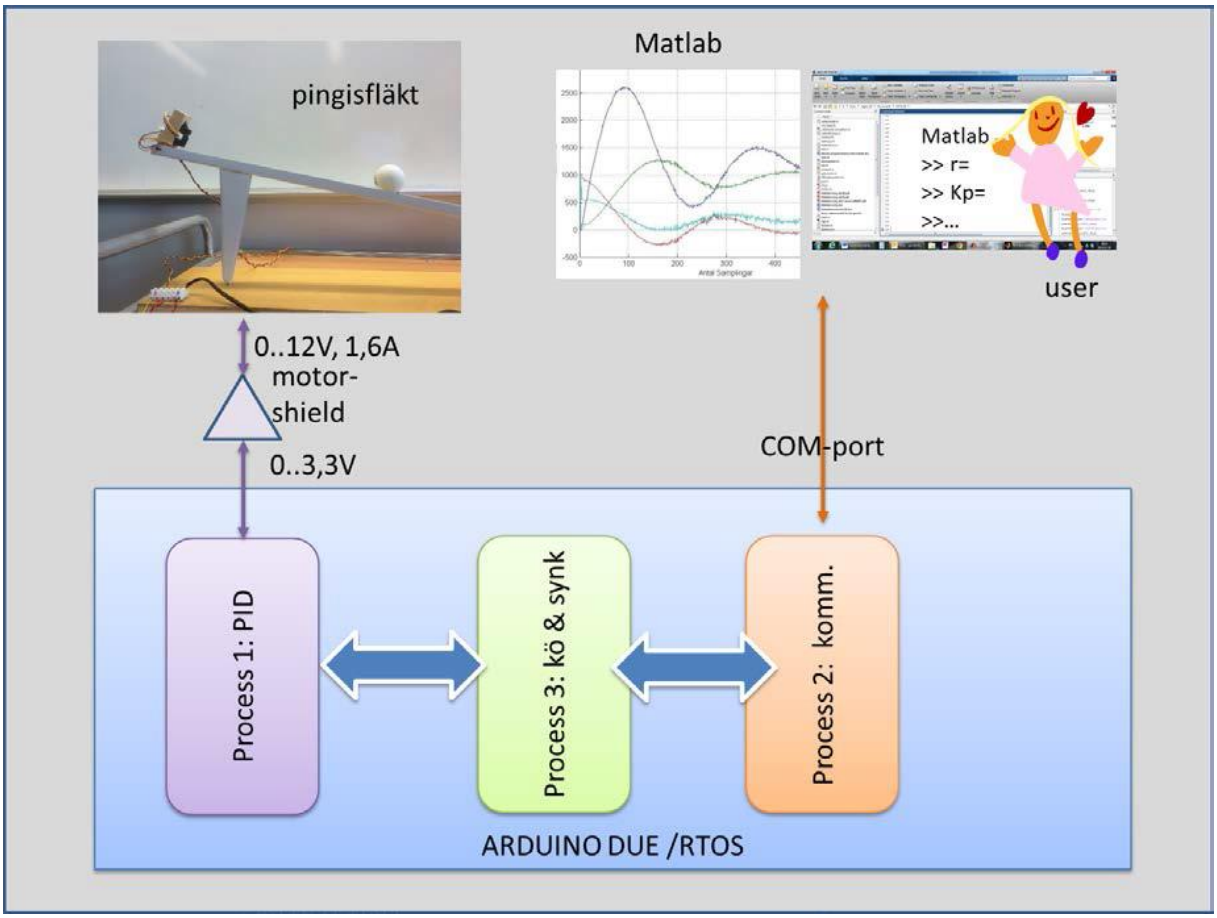
Innehållsförteckning

1. Inledning

Denna examinationsuppgift har till syfte att visa att eleven, jag, kan tillämpa olika regleralgoritmer på ett fysikaliskt system i praktik.

Systemet består av en distans sensor, som känner av pingisbollens avstånd på rälsen och en fläkt som blåser pingisbollen in i position. Dessa två mekanismer ska PID-regleras och för att uppnå ett slutresultat där pingisbollen håller sig i förvalt position måste man inkludera ett antal andra delsystem:

- De olika processerna ska schemaläggas i en RTOS.
- Läsa av värdet från distans sensorn genom ADCn på Arduino Due kortet.
- PWM styra fläkten genom PWM kanalen på Arduino Due kortet.
- Reglera systemet med hjälp av PID så att det önskade positionen på pingisbollen uppnås autonom.
- Uppnå tvåvägs kommunikation mellan Matlab och Arduino Due kortet.
- Koppla ihop de olika komponenterna, dvs. Arduino Due kortet, Motor Shield, powerboxen och systemet.



Figur 1:Representation av uppgiften[2]

2. Syfte

Syftet med denna rapport är att visa hur de olika problemen hos prototypen har lösts, samt integrera de olika delsystemen i ett fungerande helsystem.

3. Systemdelar

3.1. RTOS

RTOS som användes var FreeRTOS som kommer som en extension i Atmel Studio(libraryn finns även att ladda hem separat). Denna RTOS:en möjliggjorde schemaläggningen av de olika processerna så att de kunde köras simultant.

3.2. Analog-to-Digital Converter(ADC)

Arduino Dues ADC är den som användes. Den hade till uppgift att läsa av värdet från sensorn som sitter på ena änden av rälsen och gjorde så med 12 bitars upplösning vilket gav noggrannare värde. Vi använde Pin 8 på Due kortet, Channel 10 enligt ASF.

3.3. Pulse Width Modulation (PWM)

Den interna PWMn på Due kortet användes. Denna fanns det lite problem med eftersom det fanns knappt någon information hur den används i samband med SAM3X8E, alltså processorn på Due kortet. Denna fick ”jury riggas” eftersom vi först och främst inte kunde hitta en ledig pin/kanal som inte Motor Shielden R3 använde. I slutändan använder vi Digital Pin 7 på Due kortet som kopplas med en tråd till Digital Pin 3 på Motor Shielden. Själva Motor Shieldens Pin 3 är urkopplad ur Due kortet.

PWM Clock A är satt till 1Mhz och Master Clock till 84Mhz.

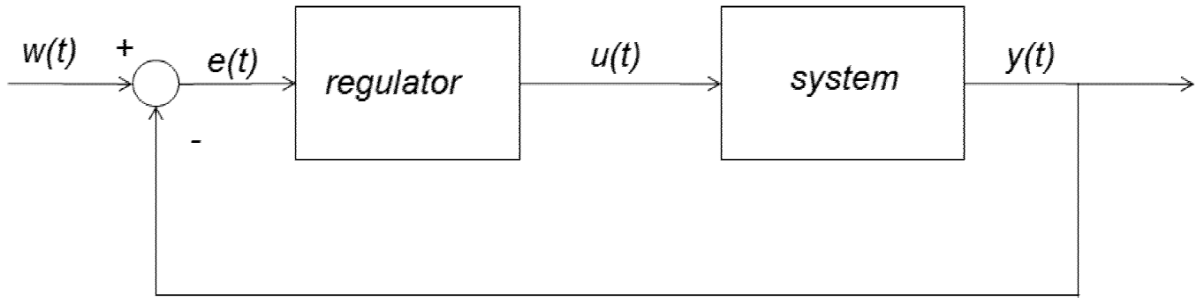
Denna används till att PWM styra motorn till fläkten enligt PID-regleringens utvärde.

3.4. Motor Shield R3

För att skydda Due kortet från den höga spänningen användes en Motor Shield R3[4]. Som skrivet ovan, använder den några av Dues pinnar för att fungera, i vårt fall Pin 12 till direction setting, Pin 9 till broms (vi använde inte denna, den var alltid satt till 0) och Pin 3 används till att styra motorn genom kanal A på skölden.

3.5. Reglering

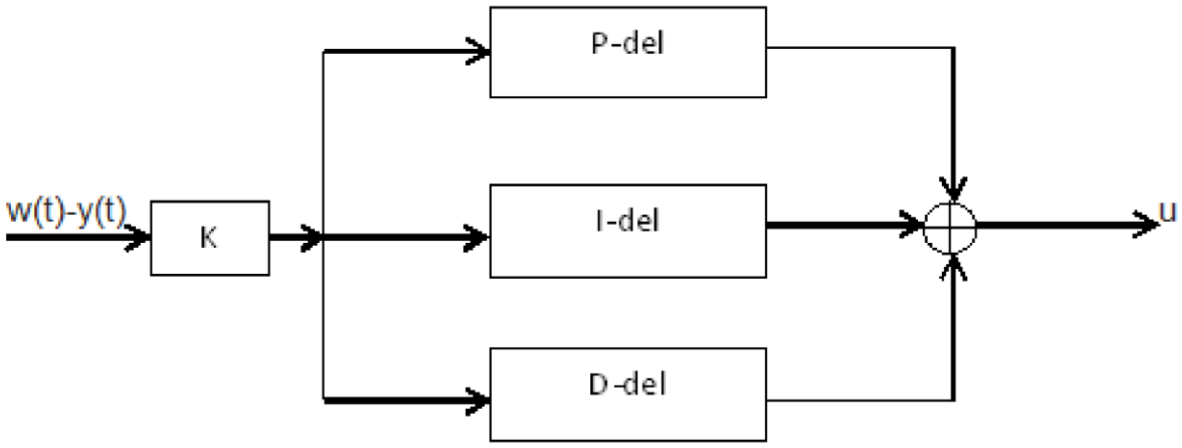
Reglerteknik är en lära som man använder när man vill göra ett tekniskt system autonom, så att den själv uppfyller kraven användaren ställer på den. Den gör så genom att kontrollera en eller flera storheter och att regulera den kräver en regulator[1]. Där i sker regleringen, som nedanstående blockdiagrammet visar. w(t) står för börvärdet som kommer igenom regulatorn och in i systemet, för att sedan kommer ut som ärvärdet som är y(t) på blockdiagrammet. Detta värde återkommer in och jämförs med börvärdet som skillnad som ger felvärdet märkt e(t) = w(t) – y(t) på blockdiagrammet. Regulatorn ger ut u(t) vilket är styrsignalen och denna påverkar systemets utsignal.



Figur 2: Blockdiagram av reglerkrets[3]

Eftersom jag nyligen lämnade in Gions labbar som jag blev godkänd på, så hade jag kunskapen fräscht i minnet vilket hjälpte med regleringen.

Det som skulle regleras i denna uppgift är pingisbollens position, och efter att ha kommit fram till slutsatsen att PID-regulator fungerade bäst i lab3c så valdes den även här. Ett blockdiagram på hur en PID-regulator, används mest i industrin[2], ser ut kan ses nedan:



Figur 3: Blockdiagram för en PID-regulator

3.5.1. P-reglering

P-reglering där styrvärdet är proportionell mot (e) alltså felvärdet. Denna justeras genom att man multiplicerar den med konstanten Kp, som är den proportionella förstärkningskonstant.

$$U = K_p * e, U = styrsignal; K_p = proportionella\ förstärkningskonstant; e = felvärdet$$

När den proportionella förstärkningskonstanten ökar så ökar även snabbheten, men stabiliteten minskar. Enligt teori kan man inte lösa problemet med endast P-reglering eftersom systemet behöver ett styrvärde som inte är noll. Detta innebär att när bollen är på plats, alltså felvärdet blir 0, så blir även styrsignalen 0.

3.5.2. I-reglering

I-reglering där utsignalen är integral av felvärdet e. Eftersom integralen är en viss utsignal i en viss tidpunkt, så är tiden väldigt viktig i regulatorn och den beror på felvärdet i just den tidpunkten.

Genom att kombinera P-reglering och I-reglering, kan vi utnyttja P-regulatorns goda egenskaper och bli av med problemet som sker då styrvärdet blir 0 eftersom I-regulatorn inte tillåter det.

När man minskar den integrala konstanten K_i så får man en bättre elimination av de kvarstående felen.

3.5.3. D-reglering

D-reglering där regleringen är beroende av derivatan av insignalen. Detta innebär att när derivatan är skild från 0, blir även utsignalen från D-regulatorn det. Men utsignalen blir 0 när felvärdet blir konstant.

3.5.4. PID, tumregler

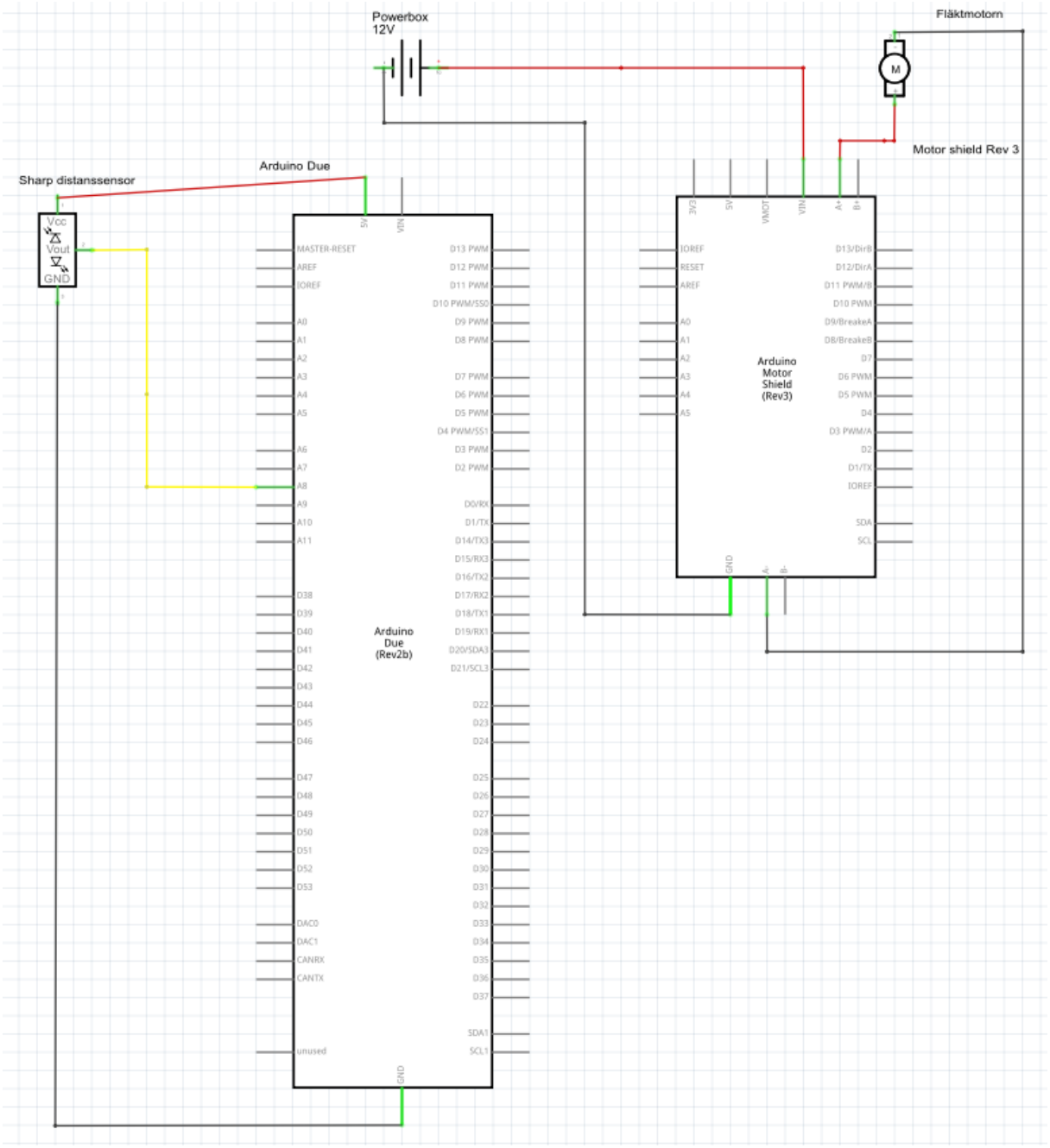
Precis som i lab3c, användes Ziegler-Nichols svängningsmetod för att ställa in PID-regulatorn. Tumreglerna för att ställa in den är följande:

- PID-regulatorn ställs först in som en P-regulator
- Man ökar den proportionella konstanten K_p fram tills $K_p=K_0$ då systemet börjar oscillera.
- Efteråt mäter man T_0, periodtiden för oscillationen.

3.6. Kommunikation(Matlab)

Denna problemet löstes aldrig. Den tog mest tid, och efter många försök så fick jag inte Matlab att plotta värdena eftersom jag aldrig uppnådde någon form av kommunikation mellan Matlab och Due kortet med koden uppladdad.

4. Koppling



Kretsschemat för uppgiften. Vill påpeka att Motor shielden sitter på Due kortet och är ställd parallellt med de pinnarna den skulle suttit på.

5. Resultat

Som skrivet ovan, är kommunikationen mellan Matlab och Due kortet det enda olösta problemet. Denna visar sig vara kritisk eftersom PID-regulatorn inte kan ställas in genom Matlab och jag får inga plottar att visa upp för själva regleringen.

6. Diskussion

Som jag nämnde tidigare, var det det största problemet Matlab. Den enda olösta delen av uppgiften som visade sig vara kritisk.

Regleringen gick smidigt, eftersom jag hade redan skrivit de matematiska formlerna för föregående examinationsuppgift med vattentankarna och det fanns inga större problem att skriva om koden för denna uppgift.

RTOS var det likaså inga problem med, den har använts i tidigare examinationsuppgift såsom projektet för ingenjörer.

Även ADCn kunde jag ta från tidigare labbar och bara anpassa det till denna uppgift. Det fanns inga konstigheter här, och eftersom det var en 12bitars upplösning var värdena som lästes av sensorn väldigt noggranna.

PWM däremot var ett jobb att göra. Eftersom det finns jätte lite information på nätet om hur den används var vi tvungna att gräva djupare i .h filerna i ASF. Eftersom jag pluggade Middleware programming som fristående kurs, var detta inte lika skrämmande som det hade varit innan, men det var ändå väldigt jobbigt att få den att fungera och det slutade med att den rätta kanalen inte hittades så Pin 7 fick användas och ”jury riggas” till Pin 3 på shielden.

Vill även lägga till att ett märkligt fel, händer med koden min grupp har skrivit. Den vägrar att ”builda” om man använder *Build* knappen, men *Rebuild* fungerar utmärkt.

7. Källförteckning

[1] B. Thomas, Modern Reglerteknik.

[2] G.K. Svedberg, Examinationsuppgift 2014

[3] G.K. Svedberg, ”Lab3c”

[4][Online] https://www.arduino.cc/en/Main/ArduinoMotorShieldR3