Cubli mini-project

Anteckningar

**Distribution**

Combine: Simon Yngve

[1 Introduktion 3](#_Toc456364955)

[1.1 Använd mjukvara 3](#_Toc456364956)

[2 Definitioner 4](#_Toc456364957)

[2.1 Koordinatsystem 4](#_Toc456364958)

[2.2 Referensramar 4](#_Toc456364959)

[3 Kommentarer på rapporten 5](#_Toc456364960)

[4 Simulinkmodeller 5](#_Toc456364961)

[4.1 Fixa dependencies 5](#_Toc456364962)

[4.1.1 Waijung 5](#_Toc456364963)

[4.1.2 UC3M 6](#_Toc456364964)

[4.1.3 Embedded Coder 6](#_Toc456364965)

[4.2 Kör Simulinkmodell 7](#_Toc456364966)

[4.2.1 Övriga anteckningar – detta behöver förmodligen inte göras 8](#_Toc456364967)

[5 Arbeta med Simulinkbibliotek 9](#_Toc456364968)

[6 Batterier 9](#_Toc456364969)

[7 IMU 9](#_Toc456364970)

[8 Komplementärt filter 9](#_Toc456364971)

[9 Filer 10](#_Toc456364972)

[10 Att göra 10](#_Toc456364973)

[11 Anteckningar 11](#_Toc456364974)

[11.1 Utvecklingsanteckningar 11](#_Toc456364975)

# Introduktion

Detta dokument är anteckningar från mitt arbete med kuben.

Det jag har hunnit göra är

* Instruktioner om hur man hanterar kuben och modeller
* Flyttat IMU:n
* Skrivit om det komplementära filtret till att hantera både bias och skalfaktorer
* Påbörjat en simuleringsmodell
* Skrivit nya körinstruktioner

## Använd mjukvara

Matlab2015b med Embedded Coder

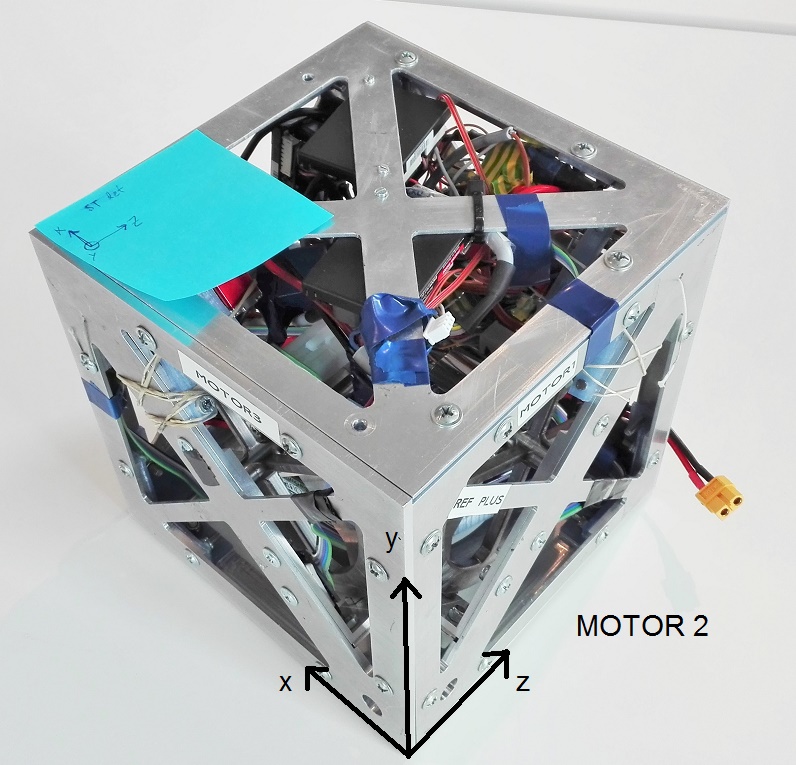
Se även kapitel 4.1.

# Definitioner

## Koordinatsystem

Oklar definition från exjobbet. Definierar koordinatsystemet så att motor 1 roterar kring x-axeln, motor 2 kring y-axeln och motor 3 kring z-axeln, på så sätt att koordinataxeln pekar från svänghjulet igenom kuben. Origo väljs till det hörn alla motorsidor har gemensamt (vilket också är det hörn som kuben ska balansera på).

Mätningar har gjorts i detta koordinatsystem.



Figur 1: Valt koordinatsystem. Origo är placerat i nedersta hörnet.

Rekommendation är att använda sig av standarddefinition av Eulervinklar alternativt quaternioner. I dagsläget definieras kubvinkeln som positiv rotation kring beskriven z-axel.

## Referensramar

Jag har räknat med att inertiell referensram och jordreferensram är samma, dvs vi bortser ifrån jordrotationen.

Inertiell ram: Index: i. ”Underlaget”

Kroppsram (body): Index: b. Kuben.

Sensorram: Index: s. IMU.

Anger beskrivning mellan referensramar enligt:

Alltså, hastigheten (vektoriell) uttryckt i sensorram fås genom att multiplicera hastigheten i inertiell ram med tranformationsmatrisen .

# Kommentarer på rapporten

Ekvation 2.3: Borde man ta hänsyn till att COG har en hastighet, inte bara att kroppen roterar? Det går att bortse ifrån hastighetstermen, men då måste tröghetsmomentet anges kring den stillastående punkten.

Ekvation 2.5: Borde det vara I\_f\*\ddot{\theta}\_c + I\_w(\ddot{\theta}\_w**+\ddot{\theta}\_c**)? Oftast är ju dock abs(\ddot{\theta}\_c) << abs(\ddot{\theta}\_w), men kan man bortse ifrån det? Alternativet är att låta I\_f innehålla I\_w. I nuläget är alltså I\_f för litet.

Angående jump up: Schweiziska studien har en pinne som gör att hjulet tvärstannar. Varför har man valt cykelbroms?

Det saknas modellering av de borstlösa motorerna som används i simuleringsmodellen. Krävs det en bättre modell än att bara översätta spänning till moment linjärt (T=Ku)? Jag antar att den dynamiken är väldigt snabb.

# Simulinkmodeller

När detta kapitel skrevs användes Björn och Eriks modeller som ligger i CODE.

## Fixa dependencies

Om en Simulink modell öppnas med en ny installation av Matlab kommer vissa referenser att saknas. Två blocksets saknas för Simulink: Waijung och UC3M. Följande sektioner beskriver hur de installeras.

Detta är blandat anteckningar med instruktioner för att jag inte vet exakt vad som behövs.

### Waijung

Laddar ner ifrån <https://www.aimagin.com/download/>

Lägg filerna i mappen där du vill att Waijung ska ligga.

Kör install\_waijung.m

Waijung saknar ST Link. Fortsätt utan att installera.

Hittar inte BC\_USB. Kan det vara ST Link?

#### ST Link – oklar betydelse

Installera ST Link via länken som anges under Matlabinstallationen: <http://www.st.com/web/en/catalog/tools/PF258168>

Den pekar till en annan sida:

<http://www.st.com/content/st_com/en/products/embedded-software/development-tool-software/stsw-link004.html>

Waijung saknar fortfarande ST Link. Det är därför oklart om detta behövs. Skulle kunna täckas av UC3M-förfarandet.

### UC3M

Frågar Erik och Björn. Svar från Erik 2016-07-01:

"

Du behöver också UC3M Addon Blockset. Vilket är ett andra blocksett som vi använder just för att göra usb kommunikationen lättare.  
 Jag tror de var denna länk som vi ladda ner det på: <https://www.aimagin.com/forum/viewtopic.php?f=2&t=269>

För övrigt är den hemsidan mycket användbar vad det gäller de blocksettet, eftersom det är bara där de finns information om det.

"

Installationsanvisningar finns på sidan eller i en av rar-filerna.

Gå till sidan, laddar ner filerna:

<https://www.aimagin.com/forum/download/file.php?id=507>

<https://www.aimagin.com/forum/download/file.php?id=466>

<https://www.aimagin.com/forum/download/file.php?id=374>

&-tecknet (%26) gillas inte av OneDrive. Ta bort.

Följ installationsinstruktionerna ovan, använd v2.6.

Använd ej v2.4.

Installera innehållet från ”STM32 USB Com Port Drivers.rar”

Inga referenser bör nu saknas!

Testa att bygga 2016-04-19-UsedAtMatlabExpo/OnBoardOk.slx. Inga fel, 14 varningar.

### Embedded Coder

Toolboxen Embedded Coder krävs för att programmera och läsa data från kuben.

Simulink→Tools→Run on target hardware→Install/Update support package

Välj ST Microelectronics → ST32F4

Gå igenom installationen.

Behöver ladda ner CMSIS från ARMs webbplats. Behöver skapa konto.

Leta upp CMSIS-mappen i Matlab.

## Kör Simulinkmodell

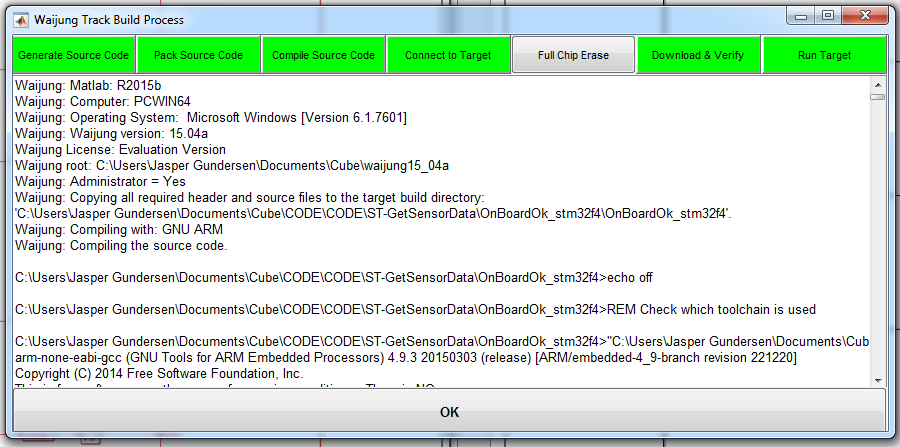
Möte med Erik 12/7 2016:

Det finns två typer av modeller: en för kod som körs på kuben, och en typ som körs på datorn. Den typ som körs på datorn hämtar data från kuben, t.ex. för att plotta IMU-data.

ST-kortet har två USB-anslutningar. Den ena anslutningen används för att programmera kortet; den andra används för att kommunicera data mellan ST-kortet och utvecklingsdatorn. Båda behöver vara inkopplade.

Man kan visa data i Simulink genom att låta kubprogrammet skicka data via USB och köra en simulinkmodell på utvecklingsdatorn som läser från USB.

Tutorial för att programmera kuben och se data på utvecklingsdatorn:

1. Anslut båda USB-sladdarna från STM-kortet till utvecklingsdatorn
2. Gå till mappen work/Stefan/code/codegeneration
3. Öppna OnBoardMatlabExpo2.slx  
   USB VCP Send är det block som skickar data via USB  
   Den nedersta delen av modellen tillhör den motor som körs för att balansera på kanten
4. Uppdatera diagram tre gånger (Ctrl+D) (Detta för att Waijung består av tre lager)  
   Tänk på att current directory i Matlab måste vara den där simulinkmodellen ligger.
5. Bygg koden till kuben (Build model C:\Users\Stefan\Documents\Mina mottagna filer\E7E2F9F9.PNG, Ctrl+B). Det kommer upp en ruta med gröna block som tänds.  
     
   När det är grönt till höger körs koden på STM-kortet.
6. Öppna OnBoardMatlabExpo2Computer.slx
7. Se till att USB-receiver-blocket är inställt med samma parametrar som USB-send-blocket i OnBoardMatlabExpo2. Maximalt 9 parametrar kan skickas (om jag kommer ihåg rätt).
8. Välj COM-port tillhörande STMicroelectronics Virtual COM Port. Kolla enhetshanteraren/device manager under Ports (i vårt fall hamnade den på COM4). Det är den port som data skickas från STM-kortet på. Notera att COM-porten eventuellt inte dyker upp förrän det finns en modell på STM-kortet som skickar data.
9. Kör
10. Tryck på kalibreringsknappen på kuben för att aktivera algoritmer om det behövs (pga det block som heter ”Callibration Motor 3 Buttom1”)

Motorerna kan antingen styras med en analog signal eller PWM-signal. Analog signal styr på volt (två per motor), PWM-signal styr på ampere. STM-kortet har bara tre analoga utgångar, därför har amperestyrning med PWM-signal valts. Detta ger struligare modell.

### Övriga anteckningar – detta behöver förmodligen inte göras

Öppnar 2016-04-19 \*MatlabExpo\*/OnBoardOk.slx.

Väljer External som target. Trycker play. – Lärdom: External är inte implementerat. Alla modeller hittills kör Normal (alltså inställningen till höger om simuleringslängd i tid).

Saknar stm32f4-funktioner, bl.a. stm32f4\_basicpwm\_fixed.

Lägger till alla undermappar i waijung-mappen till Matlab-pathen.

Modellen stannar efter kompilering, verkar ändå starta. 14 warnings.

Kompilatorn hittar ej sys/reent.h

Installerar Windows SDK

Kör mex -setup. Byter till Windows SDK 7.1.

Kör mex -setup. Byter tillbaka Microsoft Visual C++ 2010.

# Arbeta med Simulinkbibliotek

Tanken är att alla Simulinkdelar som återanvänds på flera ställen, framför allt huvuddatorns kod, ska ligga i Simulinkbibliotek. Jobba i biblioteken genom att jobba i modellen där de används, högerklicka och välj Library link→Disable link. När arbetet med biblioteksblocket är klart högerklickar man och väljer Library link→Resolve link. I rutan väljer man Push om man vill spara ändringarna till biblioteket. Kom ihåg att spara biblioteket.

Att lägga till block i biblioteket görs genom att kopiera ett block dit och spara biblioteket. Blocket får sedan kopieras från biblioteket till modellen den kom ifrån för att en länk ska skapas.

# Batterier

Ladda batterierna med tillhörande tre laddare. Kontrollera att blå, svart och röd kabel har kontakt hela vägen från laddare till batteri. Batterierna är färdigladdade när alla lampor på laddaren lyser grönt.

Låt aldrig batterierna ladda ur. (Läs om Li-Po-batterier)

# IMU

Det sitter tre IMU:er i kuben. Det bör bara behövas en.

Om man tittar i databladet till MPU9250 (förhoppningsvis samma som MPU 92/65 som anges på kortet) verkar det som att FS\_SEL = 0 och AFS\_SEL = 0. Dvs, för att få grader/s ska gyromätvärden delas med 131 och acceleration i g (9.81 m/s/s) ska accelerationmätvärden delas med 16384.

IMU:n satt inte perfekt alignad med kuben (den satt med y-axeln längs diagonalen på motor 3-sidan, med y-axeln pekandes bort från rotationskanten), vilket gjorde att g-vektorn inte gick i den riktning som vi tror. Då skulle man vara tvungen att uppskatta misalignment, vilket är svårt. Misalignment verkar inte gå att sära ifrån skalfaktor.

Jag fäste om IMU:n mot en kant på insidan av kuben, och sedan transformerades navigeringslösningen. Ännu bättre vore om den kunde fästas med bättre fästmedel, samt sitta så nära rotationspunkten som möjligt.

# Komplementärt filter

Filtret som används för att beräkna kubens vinkel är ett komplementärt filter. Det har parametern alfa som avgör hur mycket accelerometer- respektive gyrodata ska vägas in. Störningar är typiskt skakningar, vinkelacceleration och centripetalkraft. Se nedanstående jämförelser. Björn och Erik har använt alfa=0.995, vilket jag också har valt.

Den nya komplementära filtret skattar bias och skalfaktor för a\_x, a\_y och ω\_z. Se kod. Eftersom det är två okända parametrar per sensor behövs två fasta roteringar av kuben. Detta för att exitera tillstånd.

Förbättring: Kompensera för vinkelacceleration och centripetalkraft.

# Filer

Referens: Filer från Erik och Björns exjobb.

work/Stefan/code/codegeneration: modeller för att generera kod till STM-kortet, samt simulinkmodell att köra på datorn för att läsa data. OnBoardMatlabExpo2\* är de senaste som fungerar på kuben.

work/Stefan/code/lib: innehåller lib-filer samt Matlabfunktioner. Måste läggas till Matlab path.

work/Stefan/code/dependencies: Här hade jag tänkt att waijung ska ligga, men filnamnen blev för långa för OneDrive. Waijung laddas ner från nätet, se kapitel 4.1.

work/Stefan/code/measurements: Här har jag lagt mätningar och tillhörande kod. Kör useimudata eller useimudata2 för att se hur komplementära filtret hanterar data.

work/Stefan/code/simulation: Simuleringsmodell. Huvudmodellen är cube\_2d\_simulation\_model.slx. Fungerar i dagsläget inte som den borde. Se kapitel 10. Mappens innehåll skulle kunna behöva lite mer struktur…

work/Stefan/documents: Innehåller denna fil samt nya körinstruktioner.

# Att göra

Flytta IMU och fäst kant i kant med kanten på kuben, så långt ned som möjligt. Se till att IMU:n inte kan röra sig.

Sätt fast batteriet som har lossnat.

Kantbalansering:

* Implementera nytt kalibreringsförfarande
* Mät upp tyngdpunkt för att välja rätt referensvärde

Simuleringsmodell:

* Validera simuleringsmodellen.
* Modellera strömstyrning av motor.
* IMU-data i 3D
* Ändra alla signaler till SI (radianer!)

Undersök om det går att köra modell i external mode.

Använd hjulen för att signalera kalibreringsfaser, när hjulen ändå inte används.

# Anteckningar

Man bör vara försiktig med att välta kuben för våldsamt, eftersom IMU:n är känslig för kraftiga accelerationer. (10000 g enl datablad, oklar uppfattning om vad det innebär för släpphöjd, men det låter rätt mycket.)

Skulle man kunna få den att balansera kring vilken vinkel som helst? - Nej, gränsen är vid teoretiska gränsen (recovery angle)

Används bromsarna bara för att ställa kuben upp (Jump up)?

Går det att läsa av rotationshastighet?

Är modellen validerad?

Var har modellparametrar fåtts ifrån? Om de är tagna ifrån Cubli, går det med någon säkerhet säga att de borde vara lika?

Har varje IMU-kort endast en accelerometer och ett gyro? Nej. Det finns tre IMU:er med 3 accelerometrar och 3 gyron vardera. Jag har inte fått någon utförlig förklaring till varför.

I simuleringsmodell från Björn: Verkar vara fel avstånd till tyngdpunkt.

I kalibreringsläget skulle motorerna kunna användas som indikator på när det startar och är klart.

## Utvecklingsanteckningar

Koordinatsystem?

Kan navigeringen (tillståndsestimeringen) ha hjälp av att spetsen alltid är fast? (Om man antar det.) Det minskar antalet frihetsgrader. – Man kan kompensera för vinkelacceleration och centripetalkraft. Inflytandet av dessa minskar ju närmare spetsen som IMU:n sitter.

Går det att i balanserande läge skilja på IMU-fel och tyngdpunktsfel?

Pinkonfiguration kan fås upp om man trycker på hjälp på ett UC3M-block.

Går det att köra en modell i external mode? Så att man lättare kan läsa av signaler i Simulinkmodellen.