

# Retteblad

Retteblad til projektet, Udvikling af et positioneringssystem – ved hjælp af dead reckoning, sensorintegration og Kalman filtrering.

I underafsnit 8.2.1, er der i tabellen, byttet om på nogle kolonner (accelerometerbias er byttet med gyrobias). I det efterfølgende, er der lavet vurderinger ud fra denne tabel, hvorfor disse vurderinger ikke er 100 % korrekte. Fejlen har ikke haft indflydelse på den øvrige del af rapporten.

Derfor er det rettede underafsnit 8.2.1 vedlagt og disse sider skal dermed erstatte side 85-87 i hovedrapporten.

For at gøre det lettere at benytte det udarbejdede NS-1002 i Matlab, er der udformet et diagram som beskriver, hvorledes brugergrænsefladen åbnes i Matlab og hvilke funktioner, der benyttes for at softwaren fungerer.

Dette skal i princippet ligges på CD'en sammen med softwaren, og er tiltænkt fremtidig brug og udvikling af softwaren, så en forståelse af programmet lettere kan opnås.

Dette diagram findes umiddelbart efter underafsnit 8.2.1.

På Bilags CD'en er der i mappe ' 04 C-kode IMU400CC-100' vedlagt et eksempel på en C-kode. Denne fil virker umiddelbart ikke på CD'en hvorfor denne ligeledes er vedlagt.

I Bilag G i bilagsmappen er der i indledningen, side 51, henvist til underafsnit 5.1.1 og underafsnit 5.1.2 i hovedrapporten. Dette skal i stedet være henvisninger til hhv. underafsnit 5.2.1 og underafsnit 5.2.2.

Venlig hilsen

Gruppe 09ms1002

Carsten Bundgaard Jacobsen

Rasmus Rødgaard Riisager

Lars Vestergaard Poulsen

## 8.2.1 Kalibrering

Før IMU'en kan benyttes til positionering kræves det, at IMU'en bliver kalibreret. Dette gøres bl.a. ved at bestemme bias, som er en systematisk fejl, der kan måles og korrigeres for inden positioneringen foretages.

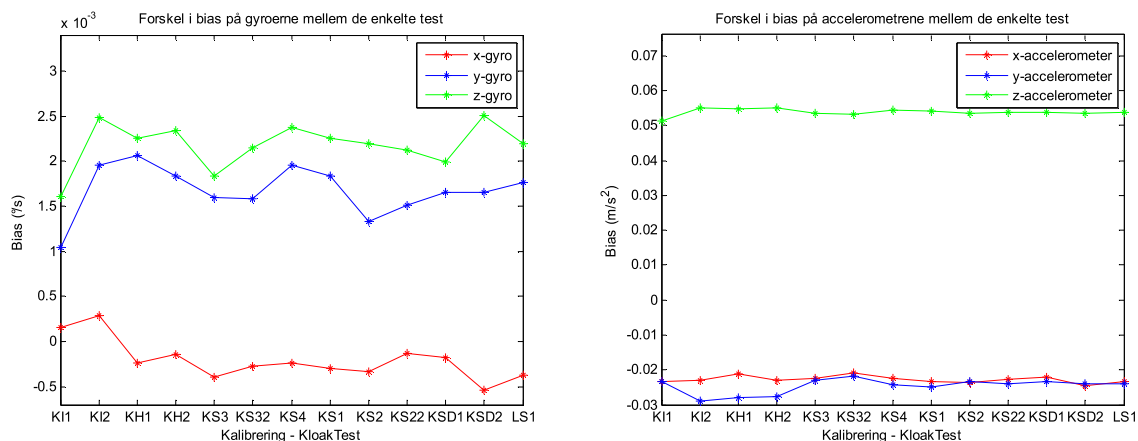
### Bias

I teorien er bias ikke stabil fra opstart til opstart, men burde være stabil, når IMU'en først er tændt, jf. Bilag C. Derfor undersøges det, om dette er tilfældet, dvs. er den stabil og systematisk, samt om den forskel der måtte være, har betydning for positioneringen af DR-IMU.

I forbindelse med hver test, er der samlet kalibreringsdata i et minut, så bias kan beregnes og der kan kompenseres for denne inden positionerne bestemmes. Derudover er der indsamlet kalibreringsdata indenfor på universitetet, både før og efter testene er udført, for at undersøge, om der sker en ændring af bias. Der ses dermed på, om der er forskel i den bias, som er beregnet for hver dataindsamling. Bias er beregnet for hver af de seks sensorer i IMU'en, dvs. de tre gyroer og de tre accelerometre, og er beregnet ud fra data indsamlet på 1 minut. I Tabel 8.2 kan forskellen i bias for de seks sensorer samt temperaturen ses, for hver af de 13 gange der er indsamlet kalibreringsdata. I Figur 8.3 er disse ligeledes vist på graf, med bias på gyroerne til venstre og bias på accelerometrene til højre. I graferne er navnet på de enkelte test blevet forkortet, så f.eks. *KloakStikDrej1* hedder *KSD1*.

Test	Gyro – bias			Accelerometer – bias			Temperatur (°)
	X (°/s)	Y (°/s)	Z (°/s)	X (m/s <sup>2</sup> )	Y (m/s <sup>2</sup> )	Z (m/s <sup>2</sup> )	
<i>KloakInde1</i>	0,00016	0,00105	0,00161	-0,02320	-0,02329	0,05146	27
<i>KloakInde2</i>	0,00029	0,00196	0,00248	-0,02310	-0,02879	0,05507	15
<i>KloakHoved1</i>	-0,00024	0,00206	0,00225	-0,02131	-0,02792	0,05476	14
<i>KloakHoved2</i>	-0,00014	0,00183	0,00234	-0,02288	-0,02759	0,05502	14
<i>KloakStik3</i>	-0,00040	0,00159	0,00184	-0,02252	-0,02291	0,05349	20
<i>KloakStik32</i>	-0,00028	0,00158	0,00215	-0,02074	-0,02183	0,05312	20
<i>KloakStik4</i>	-0,00024	0,00195	0,00237	-0,02226	-0,02440	0,05450	15
<i>KloakStik1</i>	-0,00030	0,00184	0,00226	-0,02322	-0,02479	0,05409	16
<i>KloakStik2</i>	-0,00034	0,00133	0,00219	-0,02360	-0,02328	0,05363	19
<i>KloakStik22</i>	-0,00013	0,00151	0,00212	-0,02280	-0,02393	0,05391	19
<i>KloakStikDrej1</i>	-0,00018	0,00165	0,00199	-0,02220	-0,02339	0,05391	19
<i>KloakStikDrej2</i>	-0,00054	0,00166	0,00251	-0,02448	-0,02386	0,05365	18
<i>Lukkesum1</i>	-0,00037	0,00176	0,00219	-0,02335	-0,02392	0,05373	16

Tabel 8.2: Beregnet bias for de forskellige tests samt temperaturen.



Figur 8.3: Forskel i bias mellem de enkelte test på hhv. gyroerne og accelerometrene. (Forskel i skala på de viste grafer)

Jf. underafsnit 2.3.4 er bias på gyroerne anført til at være  $<\pm 1,0$  °/s, samt  $<\pm 8,5$  mG  $\approx <\pm 0,0833$  m/s<sup>2</sup> for accelerometrene. Dermed kan det, ud fra værdierne i tabellen og graferne, ses, at IMU'en overholder specifikationerne.

Derudover er det umiddelbart vanskeligt, at vurdere noget ud fra værdierne i tabellen og graferne, men det ses, at bias ikke er stabil efter opstart, da IMU'en ikke har været slukket mellem testene. Det ses desuden, at bias på gyroerne fra *KloakInde1* er ca. halv så store som bias fra *KloakInde2*. Det kunne skyldes temperaturen, da denne jf. Bilag C, har indflydelse på størrelsen af bias. Temperaturen for *KloakInde1* er næsten dobbelt så høj i forhold til *KloakInde2* og generelt er bias på gyroerne fra *KloakInde1* mindre end de øvrige bias. For bias på accelerometrene, er den samme vurdering ikke mulig at lave.

I tabellen og graferne er det tydeligt at fortegnet for bias på de enkelte sensorer stort set er ens. F.eks. er bias for X- og Y-accelerometrene begge negative, mens bias for Z-accelerometeret og Y- og Z-gyroerne er positive. Mht. bias på X-gyro, så er de to *KloakInde* tests positive mens de øvrige er negative. Derfor er det heller ikke muligt at sige, at bias systematisk er positiv eller negativ på den enkelte sensor.

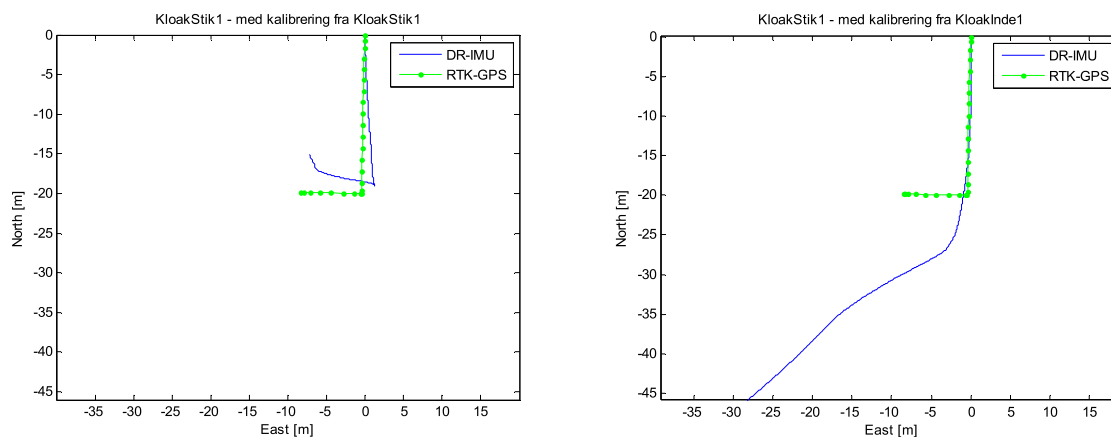
Mht. de specifikke værdier, kan det være vanskeligt at vurdere, hvor stor indflydelse det har på positioneringen, når der f.eks. er en forskel på hhv.  $-0,00009$  m/s<sup>2</sup>,  $0,00550$  m/s<sup>2</sup> og  $-0,00361$  m/s<sup>2</sup>, i bias for X-, Y- og Z-accelerometrene ved data fra hhv. *KloakInde1* og *KloakInde2*.

For at give et billede af bias betydning eller forskellen på at bruge de forskellige bias, gennemføres et beregningseksempel for betydningen. Dette gøres ved at foretage en positionsbestemmelse ud fra forskellen mellem bias i *KloakInde1* og *KloakInde2*. Funktionerne og data anvendt til denne beregning kan ses på Bilag CD.09.02. Positionsbestemmelsen giver en samlet afstand på 0,003 meter efter et sekund og 11,595 meter efter 60 sekunder, hvilket stemmer godt overens med teorien, der siger, at en fejl på accelerometer bias giver en fejl i positionen, svarende til tiden i anden. Vigtigheden af, at anvende den korrekte kalibrering er dermed tydelig. Det er samtidig klart, at en forkert bias, hurtigt vil give store fejl i positionsbestemmelsen.

Det har ikke været muligt at udarbejde et lignende eksempel for gyroerne, men for at vise kalibreringens betydning generelt for positioneringen, kan vises et eksempel, fra en test, som både beregnes med bias indsamlet umiddelbart inden testen og derefter med bias for en anden test.

Som eksempel er valgt testen, *KloakStik1*, og bias indsamlet i forbindelse med denne, samt bias fra *KloakInde1*. Forskellen kan ses i Figur 8.4.

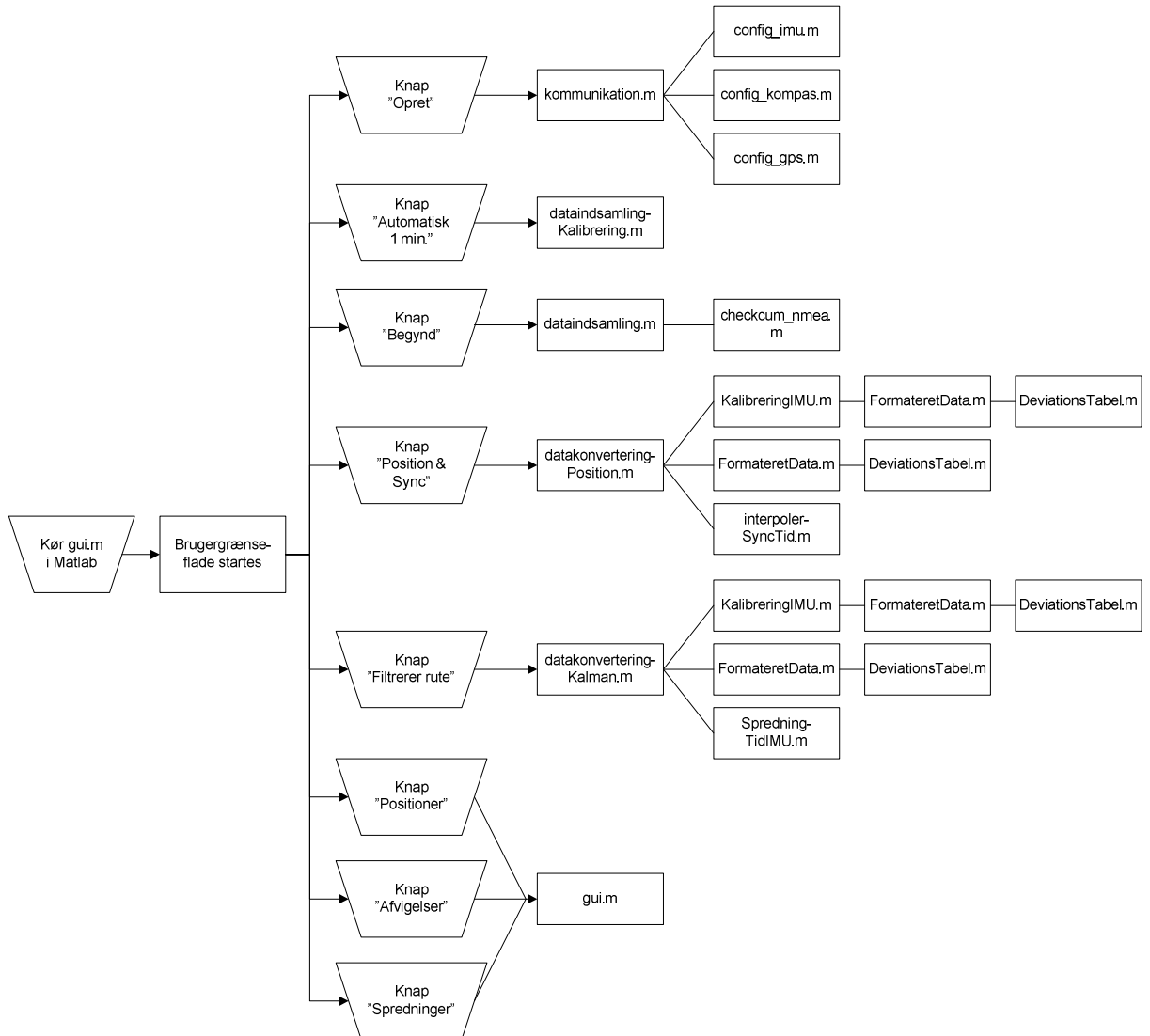
Ud fra figuren er det tydeligt, at positioneringen ikke foretages korrekt, når der benyttes andet kalibreringsdata, end det, som er indsamlet i forbindelse med testen.



Figur 8.4: Forskel i positioneringen af DR-IMU med hhv. den tilhørende kalibrering til venstre og samme positionering, med kalibreringsdata fra *KloakInde1* til højre.

På baggrund af ovenstående vurderes det derfor, at bias ikke er systematisk for IMU'en, og det er derfor nødvendigt at indsamle kalibreringsdata inden brug, også selvom IMU'en er tændt hele tiden. Da en fejl på bias giver fejl på positionerne, er det et problem, at bias tilsyneladende varierer meget, formentlig på grund af temperaturskift. Det kan derfor være nødvendigt at have en stabil temperatur i IMU'en, da det ser ud til, at temperaturen har betydning for bias. Dette kan dog ikke konkluderes endeligt ud fra de foretagne tests, men stabiliseringen af temperaturen kunne implementeres i systemet ved at placere IMU'en i f.eks. en kasse, som kan holde temperaturen stabil.

## Diagram for brug af udviklet software, NS-1002



# C-kode eksempel

```

/*****
** This is a multi-purpose data logger grab_data.cpp for Crossbow DMU
** (AHRS400, VG400, etc.) systems, and their variants. It commands the
** system into continuous mode and parsers the packets into an ASCII
** data file and onto a screen update.

** Source Code Version Label: 6170_xxxx
** Binary Version String: ADMU Data Grabber A.02

** USING grab_data
** Usage: grab_data <file> <options>
** Available options are:
** -m <mode> where <mode> is f, a, r, or c. Default is angle mode (a)
** -c <comm port> where the default is COM1 (1)
** -b <baud rate> where the default is 57600 (57600)
** -d <decimation rate> where the default is none (0)
** -t <dt> sets dt value for output (default prints counts)
** -a sets the mode to be ahrs, default is not set
** -h sets the voltage mode data to be non-ADMU (12 Bit), default is 16 Bit
** -p sets the packet to be NewPacket-ahrs (26 Byte), default is not set

** In the above options, the mode option has the following four choices:
** f:    fast raw voltage mode, currently only supported by the ADMU
** a:    angle mode
** r:    raw voltage mode
** c:    scaled sensor mode

** Given the choices above, if the mode option is angle mode, and the
** ahrs option is also set, then the system will parse roll, pitch and
** yaw (heading). The new packet option is only supported in angle
** mode, and will interpret the new packet definition.

** POSSIBLE ERRORS
** Since the program parsers the data in continuous mode, a syncing
** procedure is maintained throughout the data logging, which involves
** checking for the correct header byte, counting the bytes in the
** packet, and verifying the checksum. If the syncing is lost, a
** resynchronize procedure is begun which may hang or may take several
** hundred milliseconds to resynchronize with the continuous data flow.

** SYSTEM REQUIREMENTS
** This software will run on a PC with Windows 95, 98, or NT. It is for
** use with Crossbow standard AHRS and DMU products.

** KNOWN DEFECTS AND WORKAROUNDS
** None

*****/

#include <stdio.h>
#include <string.h>
#include <time.h>
#include <stdlib.h>
#include <time.h>
#include <errno.h>
#include <windows.h>
#include <conio.h>

#define VERSION_STRING "ADMU Data Grabber A.02\n"

#define MAX_PACKET_SIZE 255
#define MAX_BUF_SIZE 255

/**/ Define the header in the data packet ***/
#define NORMAL_HEADER 0xFF
#define NORMAL_HEADER_SIZE 1

#define EXTENDED_HEADER 0xAA55
#define EXTENDED_HEADER_SIZE 2
#define EXTENDED_HEADER_FIRST_BYTE 0xAA

/**/ Define the packet size ***/
#define HDX_RAW_PACKET_SIZE 18
#define HDX_FAST_RAW_PACKET_SIZE 16

```

```
#define HDX_COOKED_PACKET_SIZE 18
#define HDX_ANGLE_PACKET_SIZE 22

#define AHRS_RAW_PACKET_SIZE 24
#define AHRS_FAST_RAW_PACKET_SIZE 22
#define AHRS_COOKED_PACKET_SIZE 24
#define AHRS_ANGLE_PACKET_SIZE 30

#define NEW_ANGLE_PACKET_SIZE 26

/** Define the AHRS modes */
#define FAST_RAW_MODE 'f'
#define FAST_MODE_BAUD 115200
#define RAW_MODE 'r'
#define COOKED_MODE 'c'
#define ANGLE_MODE 'a'

/** Define the conversion constants */
#define VOLT_RANGE 5.0
#define MAG_VOLT_RANGE 5.0
#define TWO_EXP_TWELVE 4095
#define TWO_EXP_FIFTEEN 32767
#define TWO_EXP_SIXTEEN 65535

/** Define sensor ranges for accel in G, rate gyro in deg/sec and mag in Gauss */
double gGyroRange[] = {200.0, 200.0, 200.0};
double gAccelRange[] = {2.0, 2.0, 2.0};
double gNewPacketAccelRange[] = {10.0, 10.0, 10.0};
double gNewPacketGyroRange[] = {800.0, 800.0, 800.0};
double gMagRange[] = {1.25, 1.25, 1.25};
double gSensorVoltConversion = VOLT_RANGE/TWO_EXP_SIXTEEN;

/** Define sub-routines for the data packet request */
FILE *gErr;

#define EXIT_CHAR 'q'

static bool check_packet_valid(char mode, unsigned char* packet,
                               unsigned int packetSize, bool isAhrs,
                               bool isNewPacket);

static void get_packet(HANDLE fp, unsigned int packetSize,
                      unsigned char *packet);
static void output_packet(char mode, double dt, unsigned int packetCnt,
                          unsigned char* packet, bool isAhrs, bool isNewPacket,
                          FILE* out);
static void print_packet(char mode, unsigned char* packet, bool isAhrs,
                          bool isNewPacket);
static void restart_system(HANDLE serialPort, char mode,
                           unsigned int *packetSize, bool isAhrs,
                           bool isNewPacket);

/*****
** main
**
** Entry
** Arguments to program should include the filename to save data to and
** (optionally) the comm port to take data from
** (optionally) the mode to set the unit in
** (optionally) the decimation rate
**
** Exit
** The data gathered in the file
**
**
*****/
void main (int argc, char *argv[])
{
    char *outFileName;
    unsigned char packet[MAX_PACKET_SIZE];
    FILE *out;
    char comPort[] = "COM1";
    unsigned int baudRate = CBR_57600;
    char mode = ANGLE_MODE;
    unsigned long decimationRate=1;
    int currentArg;
    bool outputData=FALSE, badPacket=FALSE, isAhrs=FALSE, isNewPacket=FALSE;
    unsigned int packetSize=0;
    unsigned int packetCnt =0;
    unsigned int i;
    double dt=-1;
    unsigned int userChar= '0';
```

```

LPDCB serialCtrl= new DCB;
HANDLE serialPort;
COMMTIMEOUTS commTimeouts;

printf(VERSION_STRING);

if (argc < 2) { /* not enough argument */
    printf("usage: grab_data <file> <options>\n");
    printf("Available options are: \n");
    printf("-m <mode> where <mode> is f, a, r, or c. Default is angle mode (a).\n");
    printf("-c <comm port> where the default is COM1 (1)\n");
    printf("-b <baud rate> where the default is 57600 (57600)\n");
    printf("-d <decimation rate> where the default is none (0)\n");
    printf("-t <dt> sets dt value for output (default prints counts)\n");
    printf("-a sets the mode to be ahrs, default is not set\n");
    printf("-h sets the voltage mode data to be non-ADMU (12 Bit), default is 16 Bit\n");
    printf("-p sets the packet to be NewPacket-ahrs (26 Byte), default is not set\n");

    getchar();
    exit(1);
}
currentArg = 1; /* 0 is the program name */
outFileName = argv[currentArg]; currentArg++;
while (argc > currentArg) {
    switch (argv[currentArg][1]) {
        case 'm': /* mode */
            currentArg++;
            mode = argv[currentArg][0]; currentArg++;
            break;
        case 'c': /* comm port */
            currentArg++;
            comPort[4] = argv[currentArg][0]; currentArg++;
            break;
        case 'd': /* decimation rate */
            currentArg++;
            decimationRate = atoi(argv[currentArg]); currentArg++;
            break;
        case 'b': /* baud rate */
            currentArg++;
            baudRate = atoi(argv[currentArg]); currentArg++;
            break;
        case 't': /* dt */
            currentArg++;
            dt = atof(argv[currentArg]); currentArg++;
            break;
        case 'a': /* ahrs */
            currentArg++;
            isAhrs=TRUE;
            break;
        case 'h': /* 12-Bit Voltage Mode */
            currentArg++;
            gSensorVoltConversion = VOLT_RANGE/TWO_EXP_TWELVE;
            break;
        case 'p': /* NewPacket ahrs */
            currentArg++;
            isNewPacket=TRUE;
            break;

        default:
            printf("unknown argument: -%c\n",argv[currentArg][1]);
            printf("type grab_data without any parameters for help");
            break;
    } /* end switch */
} /* end while */

if(isNewPacket) {
    printf("  ROLL  PITCH    YAW  GYROX  GYROY  GYROZ  ACCELX  ACCELY  ACCELZ  BIT\n");
}
else {
    if (isAhrs)
        printf("  ROLL  PITCH    YAW  GYROX  GYROY  GYROZ  ACCELX  ACCELY  ACCELZ\n");
    else
        printf("  ROLL  PITCH  GYROX  GYROY  GYROZ  ACCELX  ACCELY  ACCELZ\n");
}

out = fopen(outFileName, "wb");
gErr = fopen("err.txt", "w");

/* open serial port */
serialPort = CreateFile(comPort,GENERIC_READ|GENERIC_WRITE, 0 /* do not share */,
    NULL, OPEN_EXISTING, FILE_ATTRIBUTE_SYSTEM, NULL);

```



```

if (serialPort == NULL) {
    printf("could not open serial port, %d\n", GetLastError());
}

if (!(SetupComm(serialPort, 0x1000, 0x1000))) {
    fprintf(gErr, "error setting up comm: %d\n", GetLastError());
}

if (!(GetCommState(serialPort, serialCtrl))) {
    fprintf(gErr, "error getting comm: %d\n", GetLastError());
}

serialCtrl->BaudRate = baudRate;
serialCtrl->Parity = NOPARITY;
serialCtrl->StopBits = ONESTOPBIT;

if (!(SetCommState(serialPort, serialCtrl))) {
    fprintf(gErr, "error setting comm: %d\n", GetLastError());
}

GetCommTimeouts(serialPort, &commTimeouts);
commTimeouts.ReadIntervalTimeout = MAXDWORD;
commTimeouts.ReadTotalTimeoutMultiplier = MAXDWORD;
commTimeouts.ReadTotalTimeoutConstant = 5;

SetCommTimeouts(serialPort, &commTimeouts);

restart_system(serialPort, mode, &packetSize, isAhrs, isNewPacket);

while(userChar != EXIT_CHAR) {
    for (i=0; i<decimationRate; i++){
        get_packet(serialPort, packetSize, packet);
        if (!check_packet_valid(mode, packet, packetSize, isAhrs, isNewPacket)){
            restart_system(serialPort, mode, &packetSize, isAhrs, isNewPacket);
            packetCnt++; /* at least one packet to restart system */
            badPacket = TRUE;
        } else badPacket = FALSE;
        packetCnt++;
        if(isNewPacket){
            if ((packetCnt%10 == 0) && !badPacket) {
                print_packet(mode, packet, isAhrs, isNewPacket);
                fprintf(gErr, "packet %d\n", packetCnt);
            }
        }
        else {
            if ((packetCnt%100 == 0) && !badPacket) {
                print_packet(mode, packet, isAhrs, isNewPacket);
                fprintf(gErr, "packet %d\n", packetCnt);
            }
        }
    }
    if (!badPacket) {
        output_packet(mode, dt, packetCnt, packet, isAhrs, isNewPacket, out);
    }
    if (kbhit()) {
        userChar=getch();
    }
}

/* end while */

/* end main */

static bool check_packet_valid(char mode, unsigned char* packet,
                               unsigned int packetSize, bool isAhrs,
                               bool isNewPacket)
{
    unsigned int i;
    unsigned int checksum = 0;
    unsigned int checkout = 0;
    bool retVal = TRUE;

    switch (mode) {
    case RAW_MODE:
        if ((packetSize != HDX_RAW_PACKET_SIZE) &&
            ((packetSize != AHRS_RAW_PACKET_SIZE))) {
            fprintf(gErr, "packet size incorrect: %d should be %d for HDX or %d for AHRS\n",
                    packetSize, HDX_RAW_PACKET_SIZE, AHRS_RAW_PACKET_SIZE);
            return FALSE;
        }
    }
}

```

```

        break;
    case FAST_RAW_MODE:
        if ((packetSize != HDX_FAST_RAW_PACKET_SIZE) &&
            ((packetSize != AHRS_FAST_RAW_PACKET_SIZE))) {
            fprintf(gErr, "packet size incorrect: %d should be %d for HDX or %d for AHRS\n",
                packetSize, HDX_FAST_RAW_PACKET_SIZE, AHRS_FAST_RAW_PACKET_SIZE);
            return FALSE;
        }
        break;

    case COOKED_MODE:
        if ((packetSize != HDX_COOKED_PACKET_SIZE) &&
            ((packetSize != AHRS_COOKED_PACKET_SIZE))) {
            fprintf(gErr, "packet size incorrect: %d should be %d for HDX or %d for AHRS\n",
                packetSize, HDX_COOKED_PACKET_SIZE, AHRS_COOKED_PACKET_SIZE);
            return FALSE;
        }
        break;
    case ANGLE_MODE:
        if(isNewPacket){
            if ( packetSize != NEW_ANGLE_PACKET_SIZE ){
                fprintf(gErr, "packet size incorrect: %d should be %d for NewPacket AHRS\n",
                    packetSize, NEW_ANGLE_PACKET_SIZE);
                return FALSE;
            }
        }
        else {
            if ((packetSize != HDX_ANGLE_PACKET_SIZE) &&
                ((packetSize != AHRS_ANGLE_PACKET_SIZE))) {
                fprintf(gErr, "packet size incorrect: %d should be %d for HDX or %d for AHRS\n",
                    packetSize, HDX_ANGLE_PACKET_SIZE, AHRS_ANGLE_PACKET_SIZE);
                return FALSE;
            }
        }
        break;
    default:
        fprintf(gErr, "\nunrecognized packet type: %c", mode);
        return FALSE;
} /* end switch */

if( isNewPacket ) {
    if (packet[0] != EXTENDED_HEADER_FIRST_BYTE) {
        fprintf(gErr, "\nunrecognized NewPacket Ahrs header byte: ");
        retVal = FALSE;
    }
}
else {
    if (packet[0] != NORMAL_HEADER) {
        fprintf(gErr, "\nunrecognized header byte: ");
        retVal = FALSE;
    }
}

if(isNewPacket) {
    for (i=2; i<packetSize-2; i++) {
        checksum += packet[i];
    }
    checkout = (unsigned int)((packet[i]<<8) + packet[i+1]);
    if (checkout != checksum) {
        fprintf(gErr, "\nbad checksum: ");
        retVal = FALSE;
    }
}
else {
    for (i=1; i<packetSize-1; i++) {
        checksum += packet[i];
    }
    if (packet[i] != (checksum & 0xFF)) {
        fprintf(gErr, "\nbad checksum: ");
        retVal = FALSE;
    }
}

if (retVal==FALSE) {
    for (i=0; i<packetSize; i++)
        fprintf(gErr, "%x ", packet[i] &0xFF);
    fprintf(gErr, "\n");
    return FALSE;
}
return TRUE;
} /* is packet valid end */

static void get_packet(HANDLE fp, unsigned int packetSize,

```

```

                                unsigned char *recvBuf)
{
    unsigned int i;
    unsigned char tmpBuf[MAX_PACKET_SIZE];
    unsigned long bytesRead, byteCnt, bytesToRead;
    unsigned long recvData =0;

    byteCnt = 0;
    while (byteCnt < packetSize) {
        bytesToRead = packetSize-byteCnt;
        ReadFile(fp, tmpBuf, bytesToRead, &bytesRead, NULL);
        for(i=0; i<bytesRead; i++)
            recvBuf[byteCnt + i] = tmpBuf[i];
        byteCnt += bytesRead;
    }
} /* end get_packet */

static void  output_packet(char mode, double dt, unsigned int packetCnt,
                                unsigned char* packet,bool isAhrs, bool isNewPacket,
                                FILE* out)
{
    unsigned int i, j;

    static double initialTime, thisTime;
    static int firstTime = 1;

    static unsigned tmp[50][25];
    static int tmpCnt=0;

    double elapsedTime;
    double accel[3], gyro[3], temp, roll, pitch, mag[3], yaw;
    unsigned int hdxTime,packetSize;
    unsigned short partNumber=0,bit=0;
    static int adjustTime =1;
    static unsigned long packetCntAdjust =0;

    if (dt < 0)
        fprintf(out, "%d ", packetCnt);
    else {
        thisTime = clock() /CLOCKS_PER_SEC;
        elapsedTime = dt*(packetCnt-packetCntAdjust) + initialTime;

        if (elapsedTime+(dt*15) < thisTime) { /* lost time jump to this time */
            packetCntAdjust = packetCnt;
            initialTime = thisTime;
            elapsedTime = thisTime;
        }

        if (firstTime) {
            initialTime = clock() /CLOCKS_PER_SEC;
            firstTime = 0;
        }

        fprintf(out, "%d %f ", packetCnt, elapsedTime);
    } /* end else dt >=0 */
    switch (mode) {
    case FAST_RAW_MODE:
        if (isAhrs) packetSize = AHRS_FAST_RAW_PACKET_SIZE;
        else packetSize = HDX_FAST_RAW_PACKET_SIZE;

        for (j=0;j<packetSize; j++) {
            fprintf(out, "%c", packet[j]);
        }

        if (tmpCnt<50) {
            for (j=0; j<packetSize; j++) tmp[tmpCnt][j] = packet[j];
            tmpCnt++;
        }
        break;

    case RAW_MODE:
        i=1;
        for (j=0; j<3; j++) {
            gyro[j] = ((unsigned short)((packet[i] << 8) + packet[i+1])) *
                gSensorVoltConversion;
            i+=2;
        }
        for (j=0; j<3; j++) {
            accel[j] = ((unsigned short)((packet[i] << 8) + packet[i+1])) *
                gSensorVoltConversion;

```

```

        i+=2;
    }
    if (isAhrs) {
        for (j=0; j<3; j++) {
            mag[j] = ((unsigned short)((packet[i] << 8) + packet[i+1])) *
                gSensorVoltConversion;
            i+=2;
        }
    }
    /* mystic incantation to get temp in degrees Celsius */
    temp = (((packet[i] << 8) + packet[i+1]) * 5.0/4096.0) - 1.375)*44.44;
    i+=2;
    hdxTime = (packet[i] <<8) + packet[i+1];
    fprintf(out, "%f %f %f ", gyro[0], gyro[1], gyro[2]);
    fprintf(out, "%f %f %f ", accel[0], accel[1], accel[2]);
    if (isAhrs) {
        fprintf(out, "%f %f %f ", mag[0], mag[1], mag[2]);
    }

    fprintf(out, "%f %d\n", temp, hdxTime);
    return;

    break;
case COOKED_MODE:
    i=1;
    for (j=0; j<3; j++) {
        gyro[j] = ((short)((packet[i] << 8) + packet[i+1])) *
            (gGyroRange[j]*1.5)/TWO_EXP_FIFTEEN;
        i+=2;
    }
    for (j=0; j<3; j++) {
        accel[j] = ((short)((packet[i] << 8) + packet[i+1])) *
            (gAccelRange[j]*1.5)/TWO_EXP_FIFTEEN;
        i+=2;
    }
    if (isAhrs) {
        for (j=0; j<3; j++) {
            mag[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                (gMagRange[j]*1.5)/TWO_EXP_FIFTEEN;
            i+=2;
        }
    }
    /* mystic incantation to get temp in degrees Celsius */
    temp = (((packet[i] << 8) + packet[i+1]) * 5.0/4096.0) - 1.375)*44.44;
    i+=2;
    hdxTime = (packet[i] <<8) + packet[i+1];
    fprintf(out, "%f %f %f ", gyro[0], gyro[1], gyro[2]);
    fprintf(out, "%f %f %f ", accel[0], accel[1], accel[2]);
    if (isAhrs) {
        fprintf(out, "%f %f %f ", mag[0], mag[1], mag[2]);
    }

    fprintf(out, "%f %d\n", temp, hdxTime);
    return;

case ANGLE_MODE:
    if (isNewPacket){
        i=2;
        roll = ((short)((packet[i] << 8) + packet[i+1])) *
            (180.0)/TWO_EXP_FIFTEEN;
        i+=2;
        pitch = ((short)((packet[i] << 8) + packet[i+1])) *
            (180.0)/TWO_EXP_FIFTEEN;
        i+=2;
        if (isAhrs) {
            yaw = ((short)((packet[i] << 8) + packet[i+1])) *
                (180.0)/TWO_EXP_FIFTEEN;
            i+=2;
        }
        for (j=0; j<3; j++) {
            gyro[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                (gNewPacketGyroRange[j]*1.5)/TWO_EXP_FIFTEEN;
            i+=2;
        }
        for (j=0; j<3; j++) {
            accel[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                (gNewPacketAccelRange[j]*1.5)/TWO_EXP_FIFTEEN;
            i+=2;
        }
        partNumber = (unsigned short)((packet[i] << 8) + packet[i+1]);
        i+=2;
        bit = (unsigned short)((packet[i] << 8) + packet[i+1]);
    }

```

```

        fprintf(out, "%f %f ", roll, pitch);
        if(isAhrs)
            fprintf(out, "%f ", yaw);
        fprintf(out, "%f %f %f ", gyro[0], gyro[1], gyro[2]);
        fprintf(out, "%f %f %f ", accel[0], accel[1], accel[2]);
        fprintf(out, "%d %d\n", partNumber, bit);
    }
    else {
        i=1;
        roll = ((short)((packet[i] << 8) + packet[i+1])) *
            (180.0)/TWO_EXP_FIFTEEN;
        i+=2;
        pitch = ((short)((packet[i] << 8) + packet[i+1])) *
            (180.0)/TWO_EXP_FIFTEEN;
        i+=2;
        if (isAhrs) {
            yaw = ((short)((packet[i] << 8) + packet[i+1])) *
                (180.0)/TWO_EXP_FIFTEEN;
            i+=2;
        }
        for (j=0; j<3; j++) {
            gyro[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                (gGyroRange[j]*1.5)/TWO_EXP_FIFTEEN;
            i+=2;
        }
        for (j=0; j<3; j++) {
            accel[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                (gAccelRange[j]*1.5)/TWO_EXP_FIFTEEN;
            i+=2;
        }
        if (isAhrs) {
            for (j=0; j<3; j++) {
                mag[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                    (gMagRange[j]*1.5)/TWO_EXP_FIFTEEN;
                i+=2;
            }
        }
        /* mystic incantation to get temp in degrees Celsius */
        temp = (((packet[i] << 8) + packet[i+1]) * 5.0/4096.0) - 1.375)*44.44;
        i+=2;
        hdxTime = (packet[i] <<8) + packet[i+1];

        fprintf(out, "%f %f ", roll, pitch);
        if(isAhrs)
            fprintf(out, "%f ", yaw);
        fprintf(out, "%f %f %f ", gyro[0], gyro[1], gyro[2]);
        fprintf(out, "%f %f %f ", accel[0], accel[1], accel[2]);
        if (isAhrs) {
            fprintf(out, "%f %f %f ", mag[0], mag[1], mag[2]);
        }
        fprintf(out, "%f %d\n", temp, hdxTime);
    }

    break;
default:
    printf("unrecognized packet type in output_packet \n");
    break;
} /* end switch */

}
static void print_packet(char mode, unsigned char* packet, bool isAhrs, bool isNewPacket)
{
    unsigned int i, j;

    double accel[3], gyro[3], temp, roll, pitch, mag[3], yaw;
    unsigned int hdxTime;
    unsigned short partNumber=0,bit=0;

    switch (mode) {
    case FAST_RAW_MODE:
    case RAW_MODE:
        i=1;
        for (j=0; j<3; j++) {
            gyro[j] = ((unsigned short)((packet[i] << 8) + packet[i+1])) *
                gSensorVoltConversion;
            i+=2;
        }
        for (j=0; j<3; j++) {

```

```

        accel[j] = ((unsigned short)((packet[i] << 8) + packet[i+1])) *
            gSensorVoltConversion;
        i+=2;
    }
    if (isAhrs) {
        for (j=0; j<3; j++) {
            mag[j] = ((unsigned short)((packet[i] << 8) + packet[i+1])) *
                gSensorVoltConversion;
            i+=2;
        }
    }
    /* mystic incantation to get temp in degrees Celsius */
    temp = (((packet[i] << 8) + packet[i+1]) * 5.0/4096.0) - 1.375)*44.44;
    i+=2;
    hdxTime = (packet[i] <<8) + packet[i+1];
    printf("%5.2f %5.2f %5.2f ", gyro[0], gyro[1], gyro[2]);
    printf("%5.2f %5.2f %5.2f ", accel[0], accel[1], accel[2]);
    if (isAhrs)
        printf("%5.2f %5.2f %5.2f ", mag[0], mag[1], mag[2]);
    printf("\r");
    return;

    break;
case COOKED_MODE:
    i=1;
    for (j=0; j<3; j++) {
        gyro[j] = ((short)((packet[i] << 8) + packet[i+1])) *
            (gGyroRange[j]*1.5)/TWO_EXP_FIFTEEN;
        i+=2;
    }
    for (j=0; j<3; j++) {
        accel[j] = ((short)((packet[i] << 8) + packet[i+1])) *
            (gAccelRange[j]*1.5)/TWO_EXP_FIFTEEN;
        i+=2;
    }
    if (isAhrs) {
        for (j=0; j<3; j++) {
            mag[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                (gMagRange[j]*1.5)/TWO_EXP_FIFTEEN;
            i+=2;
        }
    }
    /* mystic incantation to get temp in degrees Celsius */
    temp = (((packet[i] << 8) + packet[i+1]) * 5.0/4096.0) - 1.375)*44.44;
    i+=2;
    hdxTime = (packet[i] <<8) + packet[i+1];
    printf("%5.2f %5.2f %5.2f ", gyro[0], gyro[1], gyro[2]);
    printf("%5.2f %5.2f %5.2f ", accel[0], accel[1], accel[2]);
    if (isAhrs)
        printf("%5.2f %5.2f %5.2f ", mag[0], mag[1], mag[2]);
    printf("\r");

    return;

case ANGLE_MODE:

    if(isNewPacket){
        i=2;
        roll = ((short)((packet[i] << 8) + packet[i+1])) *
            (180.0)/TWO_EXP_FIFTEEN;

        i+=2;
        pitch = ((short)((packet[i] << 8) + packet[i+1])) *
            (180.0)/TWO_EXP_FIFTEEN;

        i+=2;
        if (isAhrs) {
            yaw = ((short)((packet[i] << 8) + packet[i+1])) *
                (180.0)/TWO_EXP_FIFTEEN;

            i+=2;
        }
        for (j=0; j<3; j++) {
            gyro[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                (gNewPacketGyroRange[j]*1.5)/TWO_EXP_FIFTEEN;
            i+=2;
        }
        for (j=0; j<3; j++) {
            accel[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                (gNewPacketAccelRange[j]*1.5)/TWO_EXP_FIFTEEN;
            i+=2;
        }
        partNumber = (unsigned short)((packet[i] << 8) + packet[i+1]);
        i+=2;
        bit = (unsigned short)((packet[i] << 8) + packet[i+1]);
    }

```

```

else {
    i=1;
    roll = ((short)((packet[i] << 8) + packet[i+1])) *
           (180.0)/TWO_EXP_FIFTEEN;
    i+=2;
    pitch = ((short)((packet[i] << 8) + packet[i+1])) *
           (180.0)/TWO_EXP_FIFTEEN;
    i+=2;
    if (isAhrs) {
        yaw = ((short)((packet[i] << 8) + packet[i+1])) *
              (180.0)/TWO_EXP_FIFTEEN;
        i+=2;
    }
    for (j=0; j<3; j++) {
        gyro[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                 (gGyroRange[j]*1.5)/TWO_EXP_FIFTEEN;
        i+=2;
    }
    for (j=0; j<3; j++) {
        accel[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                  (gAccelRange[j]*1.5)/TWO_EXP_FIFTEEN;
        i+=2;
    }
    if (isAhrs) {
        for (j=0; j<3; j++) {
            mag[j] = ((short)((packet[i] << 8) + packet[i+1])) *
                    (gMagRange[j]*1.5)/TWO_EXP_FIFTEEN;
            i+=2;
        }
    }
    /* mystic incantation to get temp in degrees Celsius */
    temp = (((packet[i] << 8) + packet[i+1]) * 5.0/4096.0) - 1.375)*44.44;
    i+=2;
    hdxTime = (packet[i] <<8) + packet[i+1];
}

printf("%7.2f %7.2f ", roll, pitch);
if (isAhrs)
    printf("%7.2f ", yaw);
printf("%7.2f %7.2f %7.2f ", gyro[0], gyro[1], gyro[2]);
if(isNewPacket)
    printf("%7.2f %7.2f %7.2f %4d\r", accel[0], accel[1], accel[2], bit);
else
    printf("%7.2f %7.2f %7.2f\r", accel[0], accel[1], accel[2]);
break;
default:
    printf("unrecognized packet type in output_packet \n");
    break;
} /* end switch */

}

static void restart_system(HANDLE serialPort, char mode,
                          unsigned int *retPacketSize, bool isAhrs,
                          bool isNewPacket)
{
    unsigned char buf[MAX_BUF_SIZE];
    unsigned long bytesWritten, bytesToWrite;
    unsigned int packetSize=0;
    unsigned int packetCnt =0;
    LPDCB serialCtrl= new DCB;

    /* stop continuous mode */
    bytesWritten = 1;
    while (bytesWritten > 0) {
        buf[0] = 'P';
        bytesToWrite = 1;
        WriteFile(serialPort, buf, bytesToWrite, &bytesWritten, NULL);
        Sleep(100);
        bytesToWrite = MAX_BUF_SIZE;
        ReadFile(serialPort, buf, bytesToWrite, &bytesWritten, NULL);
    }
    /* set mode, should get back the capital of the value set */
    buf[0] = 0;
    while (buf[0] != (mode - 0x20)) {
        buf[0] = mode;
        bytesToWrite = 1;
        WriteFile(serialPort, buf, bytesToWrite, &bytesWritten, NULL);
        Sleep(1);
        bytesToWrite = 1;
        ReadFile(serialPort, buf, bytesToWrite, &bytesWritten, NULL);
    }
}

```

```
}

if (mode == FAST_RAW_MODE) {
    if (!(GetCommState(serialPort, serialCtrl))) {
        fprintf(gErr, "error getting comm: %d\n", GetLastError());
    }
    serialCtrl->BaudRate = FAST_MODE_BAUD;
    serialCtrl->Parity = NOPARITY;
    serialCtrl->StopBits = ONESTOPBIT;
    if (!(SetCommState(serialPort, serialCtrl))) {
        fprintf(gErr, "error setting comm: %d\n", GetLastError());
    }
}

/* stop continuous mode */
bytesWritten = 1;
while (bytesWritten > 0) {
    buf[0] = 'P';
    bytesToWrite = 1;
    WriteFile(serialPort, buf, bytesToWrite, &bytesWritten, NULL);
    Sleep(100);
    bytesToWrite = MAX_BUF_SIZE;
    ReadFile(serialPort, buf, bytesToWrite, &bytesWritten, NULL);
}

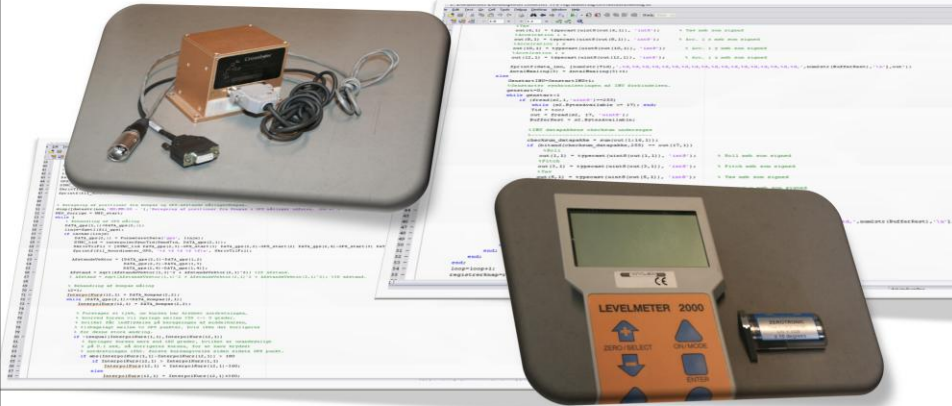
/* send a 'G' to find the size of the packet */
buf[0] = 'G';
bytesToWrite = 1;
WriteFile(serialPort, buf, bytesToWrite, &bytesWritten, NULL);
Sleep(100);
bytesToWrite = MAX_BUF_SIZE;
ReadFile(serialPort, buf, bytesToWrite, &bytesWritten, NULL);
packetSize = bytesWritten;
check_packet_valid(mode, buf, packetSize, isAhrs, isNewPacket);
/*
** now send a C to get continuous mode working
** then start recording data
*/
buf[0] = 'C';
bytesToWrite = 1;
WriteFile(serialPort, buf, bytesToWrite, &bytesWritten, NULL);

*retPacketSize = packetSize;
}
```



# Udvikling af et positioneringssystem

- Ved hjælp af dead reckoning, sensorintegration og Kalman filtrering



**Fremlæggelse af afgangprojekt**

**Tirsdag d. 23. juni 2009**

**Gruppe 09ms1002**

Jeg vil gerne sige velkommen til fremlæggelsen af vores afgangprojekt under titlen; Udvikling af et positioneringssystem – ved hjælp af dead reckoning, sensorintegration og Kalman filtrering.

## Disposition

- Velkomst og præsentation af projektet (Lars)
- Deviationstabel (Lars)
- Ny test med optimering af parametre (Rasmus)
- Bias – ustabilitet og betydning (Carsten)
- Forbedring af systemet (Carsten)

Lars

2

I de næste ca. 30 min vil vi gerne fremlægge omkring følgende emner. Undertegnet tager sig af velkomsten og en kort gennemgang af projektet. Herefter vil jeg fortælle lidt omkring Deviationstabellen, og dens indflydelse, samt præsentere resultater af nye tests af positionering med forskellige deviationstabeller, vi har udført op til fremlæggelsen. Derpå tager Rasmus over og han vil ligeledes fortælle om en ny test af systemet efter der er sket en optimering af parametrene som beskrevet i rapporten. Derefter er det Carstens tur, og han vil fortælle lidt om nye tests af bias, dens ustabilitet og indflydelse på resultaterne. Og så slutter Carsten af i form af en perspektivering.

## Udgangspunkt for projektet

- Udgangspunkt i møde med Per Aarsleff A/S, som er en entreprenørvirksomhed, med speciale i fundering og renovering af rørsystemer.
- Problemformulering:

*Hvorledes kan et dead reckoning navigationssystem, bestående af en IMU, kompas og afstande mellem GNSS positioner, designes til positionsbestemmelse i rørsystemer, og er det muligt at opnå en nøjagtighed på  $\pm 0,5$  meter eller bedre?*

Projektet har taget udgangspunkt i et møde med Per Aarsleff A/S, som er en entreprenørvirksomhed, med speciale i fundering og renovering af rørsystemer. De havde ønsket om et system til positionering af et kamera i et rørsystem, og herudfra startede projektet, og der blev udarbejdet en problemformulering som projektet har taget sit udgangspunkt i.  
(PEG PÅ PROBELMFORMU OG EVT LÆS OP)

## Udvikling af positioneringssystem

- Positioneringssystem.
  - Instrumenter
  - Positioneringsteori
  - Computer
- Simuleringskøretøj
- NS-1002
  - Brugergænseflade



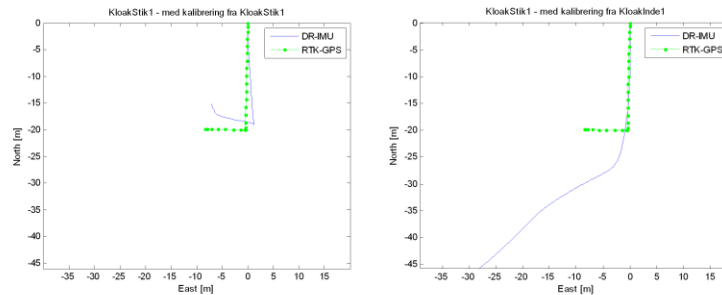
Lars

4

Ud fra problemformuleringen har vi udarbejdet et positioneringssystem. Positioneringssystemet består af tre instrumenter, IMU, kompas og GPS-modtager. Så består det af positioneringsteori, som er udarbejdet på baggrund af de benyttede instrumenter, Dead reckoning med IMU og Dead reckoning ved afstand kurs. Derudover består det af en Computer del, der benyttes som lagringsmedie og til behandling af data. For at lette brugen af systemet er der udarbejdet en brugergrænseflade i Matlab. (Åben Matlab) MATLAB: heroppe kan forbindelsen til de tre instrumenter oprettes og afbrydes. Under dataindsamling kan målesessionen navngives, der kan indsamles kalibreringsdata i et minut, og indsamling af data kan startes og stoppes. Når data er indsamlet er det muligt at få behandlet data umiddelbart herefter dvs. resultaterne kan vurderes i marken. Det sker ved at trykke på Pos & Sync. hvor positionerne til DR-AK og DR-IMU beregnes og herefter kan der foretages en Kalman filtrering. Resultaterne kan ligeledes udtegnes, positioner, afvigelse og spredninger.

# Kalibrering af instrumenter

- Kalibrering af IMU.
  - IMU observationer skal kalibreres ift. Bias, og data til beregning af bias skal indsamles umiddelbart inden brug.



Lars

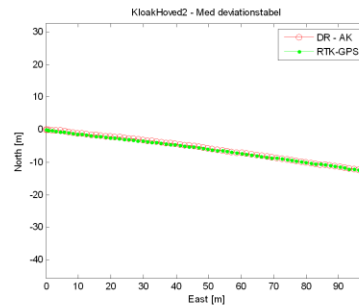
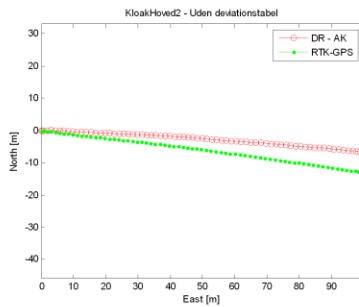
5

For at systemet kan fungere, er kalibreringen meget vigtig. Bl.a. skal IMU-observationer kalibreres i forhold til Bias og data til beregning af bias skal indsamles umiddelbart inden brug. (BESKRIVELSE AF GRAFER)

På grafer kan forskellen mellem positionering af IMU'en med kalibreringsdata indsamlet umiddelbart inden brug og kalibreringsdata indsamlet på et andet tidspunkt, ses.

# Kalibrering af instrumenter

- Kalibrering af kompas.
  - Kompasobservationer skal kalibreres vha. en deviationstabel.



Det samme er gældende for kompasset, som skal kalibreres vha. en deviationstabel. (BESKRIVELSE AF GRAFER)

På grafer kan forskellen mellem positionering med Afstand-Kurs, hvor der ikke er benyttet en deviationstabel, og til højre hvor der er benyttet en deviationstabel. Forskellen er tydelig.

## Konklusion

- DR-AK
  - Robust positionsbestemmelse – opnås gode resultater
  - Optimeres ved langsommere kørsel, samt ved at fjerne fejl fra afstandsmålingerne.
- DR-IMU
  - Høj nøjagtighed i starten – 5 til 10 sekunder. Herefter driver den meget, og der ophobes fejl.
  - Optimeres ved løbende at måle bank og elevation.
- Kalman filtrering
  - Gode resultater når der indgår position fra DR-AK
  - I de mellemliggende filtrerede punkter, er der dårlig nøjagtighed, pga. DR-IMU.

Generelt har positionsbestemmelse med DR-AK vist sig mere robust og det kan opnås gode resultater, specielt når der ikke foretages sving, og optimering kan ske ved langsommere kørsel så kompaset kan følge med, samt ved at fjerne fejl fra afstandsmålingerne når køretøjet holder stille.

Positionsbestemmelse med DR-IMU har høj nøjagtighed i starten – 5 til 10 sekunder. Herefter driver den meget, og der ophobes store fejl, hvilket resulterer i dårlig positionering over tid. Positioneringen kan optimeres ved løbende at måle bank og elevation, så der korrigeres korrekt til vandret.

I Kalman filtreringen er det, de steder hvor der indgår position fra DR-AK, muligt at opnå forholdsvis gode resultater, men i de mellemliggende filtrerede punkter som er beregnet på baggrund af IMU-observationer, er der dårlig nøjagtighed.

## Disposition

- Velkomst og præsentation af projektet (Lars)
- Deviationstabel (Lars)
- Ny test med optimering af parametre (Rasmus)
- Bias – ustabilitet og betydning (Carsten)
- Forbedring af systemet (Carsten)

Så vil jeg fortælle om deviationstabellen og dens indflydelse på kompasobservationerne og dermed positioneringen ved afstand – kurs.



## Deviationstabel

- Deviationstabellen er vigtig for at korrigere kompasobservationerne, så der kan opnås tilfredsstillende resultater.
- Ny undersøgelse med følgende formål:
  - Hvilken betydning har deviationstabellen for positionering af DR-AK?
  - Skal der laves en ny deviationstabel hver gang NS-1002 benyttes?
  - Kan der foretages færre målinger under udarbejdelse af deviationstabellen.
- Målingerne til ny deviationstabel er foretaget på en parkeringsplads i Aalborg, ca. 2 km fra det tidligere testområde.
- Der blev ikke foretaget en ny autokalibrering af kompasset.
- Målingerne blev foretaget som i projektet, og der blev taget højde for deklinationen og meridiankonvergens i forhold til KP2000J.

Lars

9

Deviationstabellen er vigtig for at korrigere kompasobservationerne, så der kan opnås tilfredsstillende resultater, som vi har set tidligere.

Derfor har vi valgt at undersøge følgende:

- Hvilken betydning har deviationstabellen for positionering af DR-AK?
- Skal der laves en ny deviationstabel hver gang NS-1002 benyttes?
- Kan der foretages færre målinger under udarbejdelse af deviationstabellen.

Vi har valgt at udarbejde en ny deviationstabel et andet sted og derefter benytte deviationstabellen

på data indsamlet i det oprindelige tests.

Målingerne til ny deviationstabel er foretaget på en parkeringsplads i Aalborg, ca. 2 km fra det tidligere testområde.

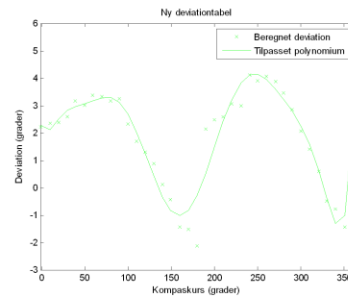
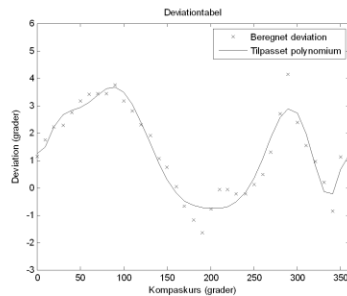
Der blev ikke foretaget en ny autokalibrering af kompasset.

Målingerne blev foretaget som beskrevet i projektet, og der blev taget højde for deklinationen og

meridiankonvergens i forhold til KP2000J.

## Ny deviationstabel

- Deviationerne plottes i forhold til den observerede kurs, og der beregnes et bedst tilpasset polynomium i forhold til punkterne.
  - 11- grads polynomium.



Lars

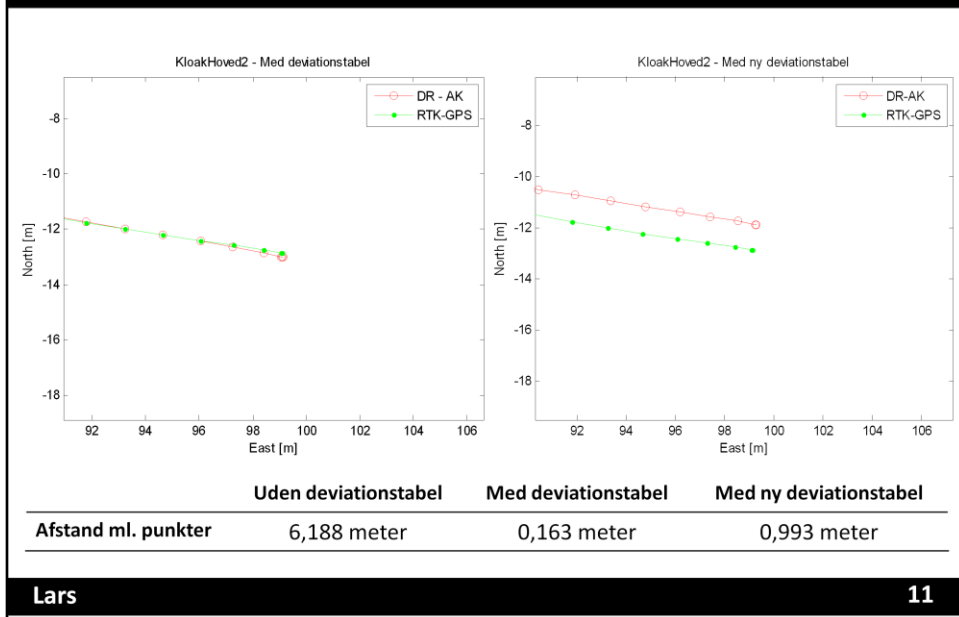
10

De beregnede deviationer plottes i forhold til den observerede kurs, og der beregnes et bedst tilpass-

set polynomium i forhold til punkterne. Igen er der valgt et 11. grads polynomium til at beskrive forløbet. (BESKRIVELSE AF GRAFER)

På graferne kan den gamle deviationstabel ses til venstre og den nye til højre. Tydeligt at der er en forskel mellem dem, specielt omkring 250-300 grader.

## Resultat med ny deviationstabel



Lars

11

Vigtigheden af en deviationstabel er stor, det så vi tidligere,

(TRYK PÅ KNAP FOR AT RETUNERE TIL SLIDE 6) Gør det tydeligt for lytter at dette sker.

Beskriv meget kort igen

(TRYK PÅ RETURKNAP FOR AT RETUNERE TIL SLIDE 11)

Den nye deviationstabel benyttes igen ved KloakHoved2 for at se betydningen af brug af ny deviat-

onstabel. (BESKRIVELSE AF GRAFER)

Svært at se den store forskel, men hvis vi zoomer ind på endepunkterne (TRYK) er forskellen tydelig.

(BESKRIVELSE AF GRAFER OG TABEL) Den nye deviationstabel giver altså et dårligere resultat, hvilket

er gældende generelt for alle testene.

Dette betyder dermed også at det vil være nødvendigt at udarbejdet en deviationstabel for hvert nyt

område der arbejdes i, selvfølgelig alt efter hvilke krav der er til resultaterne, hvorfor det også skal

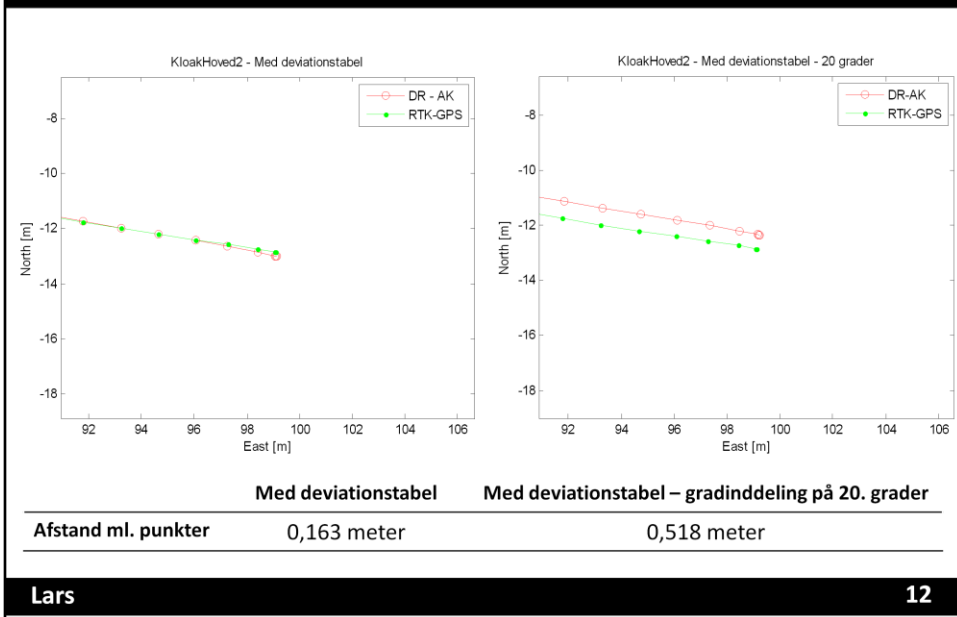
vurderes om dette er muligt og om der kunne benyttes andre instrumenter, som ville give bedre og

mere pålidelige resultater.

Ønsker man alligevel at benyttet det magnetiske kompas til positionering, vil det være en mulighed

at undersøge om processen med at udarbejde deviationstabellen kunne effektiviseres.

## Deviationstabel - effektivisering



For at vurdere om udarbejdelsen af deviationstabellen kan effektiviseres, har vi set på om de samme resultater kan opnås ved halvt så mange målinger, dvs. i stedet for måling hver 10. grad, måles der deviation ved hver 20 grad.

For at gøre dette har vi taget data fra målingerne til den første deviationstabel og fjernet hver anden kurs, og herefter udarbejdet deviationstabellen som i det tidligere. (BESKRIVELSE AF GRAFER)

Igen kan det ses at der er forskel, og benyttes den nye deviationstabel på KloakHoved2 igen, så giver

det en dårligere nøjagtighed. (BESKRIVELSE AF GRAFER OG TABEL)

Dermed skal der til den enkelte opgave hvor systemet benyttes klarlægges hvilken nøjagtighed der

ønskes og ud fra denne kan der så tages stilling til hvordan deviationstabellen skal udarbejdes.

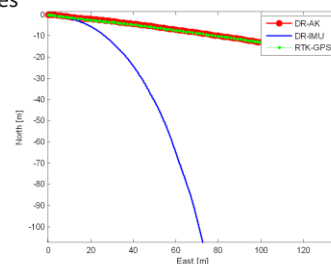
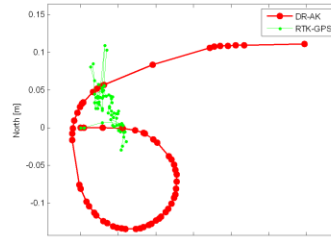
## Disposition

- Velkomst og præsentation af projektet (Lars)
- Deviationstabel (Lars)
- Ny test med optimering af parametre (Rasmus)
- Bias – ustabilitet og betydning (Carsten)
- Forbedring af systemet (Carsten)

Ja, jeg vil så fortælle om en ny test vi har foretaget for at optimere positionsbestemmelsen ud fra nogle af erfaringerne fra projektarbejdet.

## Undersøgelsens formål

- Erfaringer fra rapporten
  - RTK-GPS tilfældige fejl resulterer i konstant afstand i DR-AK
  - DR-IMU ubrugelig efter få sekunder grundet rotationsmatricen ikke opdateres
- Formål
  - Indikere at den filtrerede position kan forbedres
  - Fjerne afstande i DR-AK når vognen holder stille og foretager en drejning
  - Opdatere IMU'ens orientering, ved at beregne ny rotationsmatrice
  - Gå og dreje langsomt med vognen



Rasmus

14

En af erfaringerne fra analysearbejdet er at de tilfældige fejl på GPS positioner resulterede i at der bliver bestemt en afstand, selv når vognen holder stille.

*GRAF:*

Det ses f.eks. af denne test af DR-AK positionerne, hvor der foretages en drejning på 360 grader i samme punkt... hvor DR-AK resulterer i en position.

En anden erfaring var at positionsbestemmelsen med IMU'en reelt er ubrugelig efter nogle få sekunder, hvis ikke der inkluderes data der kan opdatere IMU'ens orientering.

En tredje vigtig erfaring var at kompasset bestemte kursen med en lavere frekvens end den hastighed vi tilbagelagde de enkelte teststruter ved.

Formålet med denne nye undersøgelse var derfor at indikere, at den filtrerede positionsbestemmelse kan forbedres, ved at:

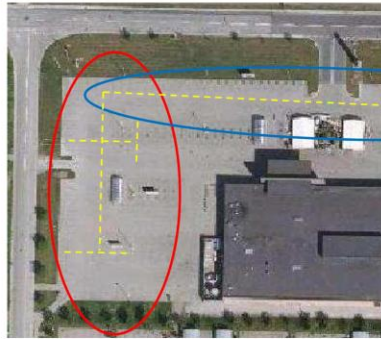
-fjerne AK afstande, hvor vi reelt holder stille.

-løbende at simulere en opdatering af IMU'ens orientering

-foretage testen i langsommere tempo, og efter en drejning først forsætte når kursen har stabiliseret sig.

## Arbejdsgang for undersøgelsen

- Testområdet og rute
  - Føtex
  - KloakStikDrej2, ca. 30 m.
- Udført søndag 14. juni. – vind 9 m/s
- Bestem elevation og bank til startpunkt
- Indsamle kalibreringsdata
- Tilbagelægge den planlagte rute
- Udvalgte hver 5. RTK-GPS punkt ( $\cong$  5 sek.)
  - Afsætte alle udvalgte punkter med vogn
  - Retningsbestemme vogn efter snor  $\cong$  heading
  - Mål elevation og bank
  - Undlod måling i udvalgte punkter, hvor der blev foretaget en drejning



Den nye test blev foretaget i samme område som testene der fremgår i rapporten, dvs. ved Føtex parkeringsplads, og som testrute blev det valgt igen at følge KloakStikDrej2, da denne har to drejninger, hvor vognen skulle holde stille. Den dag testen blev udført var det forholdsvis blæsende, med 9 m/s, hvilket kan have haft indflydelse på de resultater der er opnået.

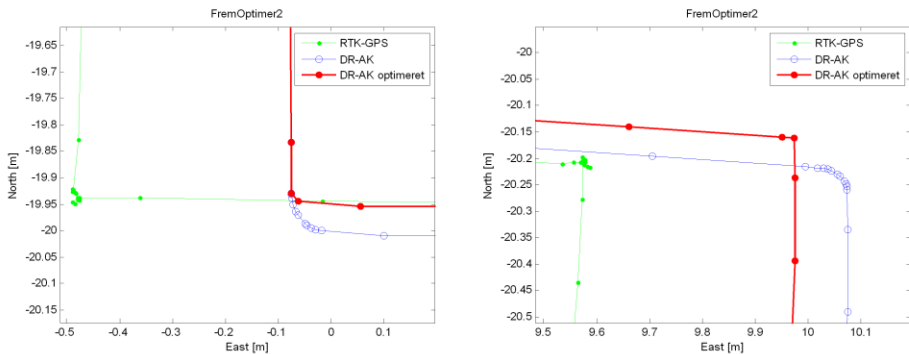
### VOGNEN:

Arbejdsmetoden for testen og de dem der er inkluderet i rapporten er følgende:

- Først opstilles vognen i startpunktet for testruten, og vognens retning flugtes med en snor der er udspændt for ruten.
- Så bestemmes elevation og bank, med inklinometeret oven på IMU'en, ved at flugte den parallelt med dens sider.
- Derefter indsamles kalibreringsdata i 1 min med brugergrænsefladen
- Og så indsamle data mens den planlagte rute tilbagelægges, ved at følge den snor der er udspændt som reference for ruten.
- Det nye ved denne test er at der herefter automatisk udvælges GPS punkter med 5 sekunders interval for den tilbagelagte rute, som overføres til GPS-modtageren.
- Disse punkter afsættes derefter med vognen, og vognen stilles parallelt med den udspændte snor.
- For hvert punkt måles elevation og bank.
- Af beregningstekniske årsager blev de udvalgte punkter hvor der blev foretaget en drejning ikke afsat.

## Resultat vedrørende DR-AK

- Middelfast når vogn foretager drejning i samme punkt
- Middelfast = 1,7 cm, på baggrund af 61 observerede afstande
- NS-1002 ændret så: Afstand  $\leq 1,7$  cm  $\rightarrow$  Afstand = 0
- Fejl i korrektion grundet blæstens indflydelse



Rasmus

16

Først kigger vi på resultaterne for DR-AK positionsbestemmelsen.

Formålet var at fjerne beregnede afstande hvor vognen reelt holder i samme punkt.

For at få et bud på hvor store afstande der skal frasorteres beregnes middelfastanden for den test vi foretog hvor vi lavede en 360 graders drejning i samme punkt.

Denne beregnede afstandsfejl beregnes til 1,7 cm

Vores program ændres herefter så afstanden sættes lig nul, hvis denne er beregnet til at være under 1,7 cm.

Vi ser lidt nærmere på hvad det betyder for den tilbagelagte rute, ved at zoome ind på de to drejninger.

For de kommende grafer er det fast at rød er optimeret bestemmelse, blå er bestemmelse med metoden fra rapporten, og grøn er den "sande" GPS rute.

### **Klik med musen**

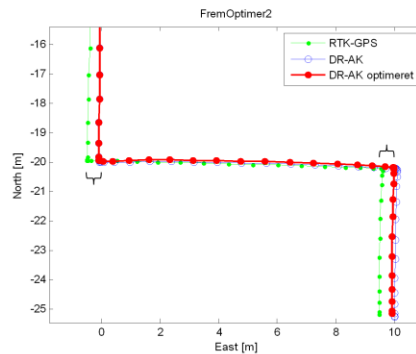
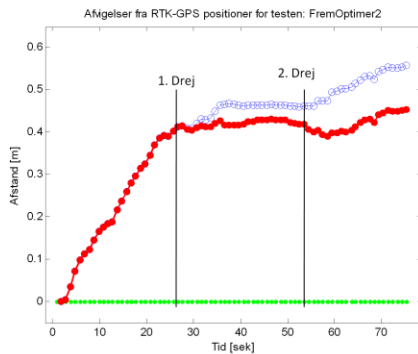
Det ses at langt de fleste fejl-afstande fjernes, men at det måske ikke er alle fejl der er frasorteret, som denne her (peg).

Det kan evt. skyldes blæsten, så vognen har vippet lidt mere end normalt under drejningen, eller en kørerfejl hvor omdrejningen ikke er foretaget præcis i samme punkt.



## Resultat vedrørende DR-AK

- Afvigelsen fra RTK-GPS falder nu ved hver drejning
- Fejl i DR-AK position må nu primært skyldes fejl i kurs



Rasmus

17

### **Venstre graf:**

Denne figur viser afvigelse til den "sande" GPS rute. Det ses at afstanden til den grønne "sande" rute stiger grundet fejl i deviationstabellen, men det ses at afvigelsen mindskes ved hver drejning, ved at afstanden mellem de to grafer bliver større.

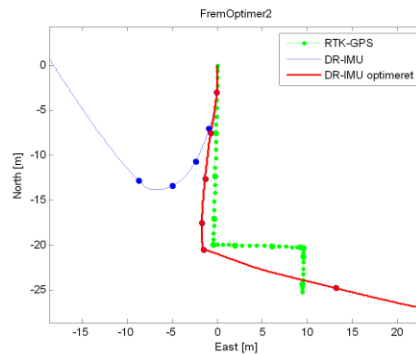
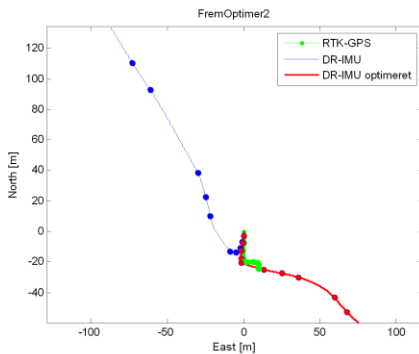
### **Højre graf:**

Af figuren til højre ses god overensstemmelse mellem den tilbagelagte afstand og den "sande", og afvigelse må nu primært skyldes fejl i bestemmelse af kursen, da disse to afstande er lige næsten store (*peg*).

Som det ses er DR-AK positionsbestemmelse altså forbedret ved denne afstandsændring.

## Resultat vedrørende DR-IMU

- Elevation og bank målt i 10 afsatte punkter (markeret)
- Rotationsmatrice opdateres, ved også at inkludere kurs som heading
- Anvendeligheden af positionsbestemmelse med DR-IMU forlænges
- Svingene er synlige, og i korrekt retning



Rasmus

18

Så går vi videre til resultaterne for positionsbestemmelse med IMU'en

Der blev målt elevation og bank i 10 af de afsatte punkter. De afsatte punkter er illustreret for hver af de tre ruter, med henholdsvis markeringerne med blå, rød og mørkegrøn. Som sagt er der ikke afsat punkter der svarer til tidspunktet der blev foretaget en drejning, da vognens retning er svær at efterligne her.

Vores program er desuden opdateret så der beregnes en ny rotationsmatrice i disse punkter ved, foruden elevation og bank, også at benytte middelkursen fra kompasset som heading.

### **Grafer:**

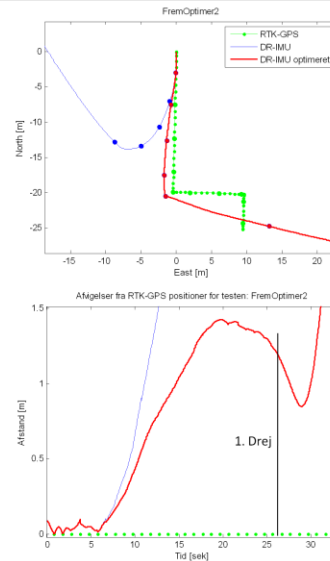
Det fremgår tydeligt at den optimerede IMU-rute stemmer bedre overens med den "sande" rute. Som det ses stemmer afstanden ned til sving 1 bedre overens, og det er nu muligt at se korrekte drejninger, om end drejning to er svag visuelt, som det ses her. Men det er en tydelig forbedring i forhold til den blå rute.

### **Højre graf:**

Man kan undre sig over at denne afstand er så stor, men dette skyldes drejningen og afstanden mellem disse to punkter svarer derfor til 15 sekunder frem for 5 sekunder.

## Resultat vedrørende DR-IMU

- Synlig forbedret positioner efter 5. sek.
- Første 30 sek. holdes afvigelsen <1,5 m.
- Hvorfor er afvigelsen mindre efter første drejning?
- Korrigere kun for den systematiske fejl, Bias
- Det antages, at elevation og bank ikke er behæftet med fejl
  - Vognen vibrerer under kørsel  
→ hældning afsat punkt ikke korrekt
  - Blæsten gav udslag på inklinometeret



Rasmus

19

### **Øverste graf:**

Denne graf viser at elevation og bank er med til at minimere afvigelsen fra den grønne "sande" rute i over en længere periode end hvis de ikke inkluderes. Hermed kan det konkluderes at anvendeligheden af IMU'ens positionsbestemmelse forlænges når dens orientering opdateres løbende.

### **Nederste graf:**

Denne figur er et udsnit af de første 30 sekunder af den øverste figur. Det ses at afvigelserne holder sig under 1,5 m, mens den efter 30 sekunder er på 18 m med den gamle metode.

### **Klik med musen**

Grunden til at afvigelsen falder her, skyldes at afstanden til den "sande" rute i en kort periode mindskes, som det ses her, hvorefter stiger igen som det ses her.

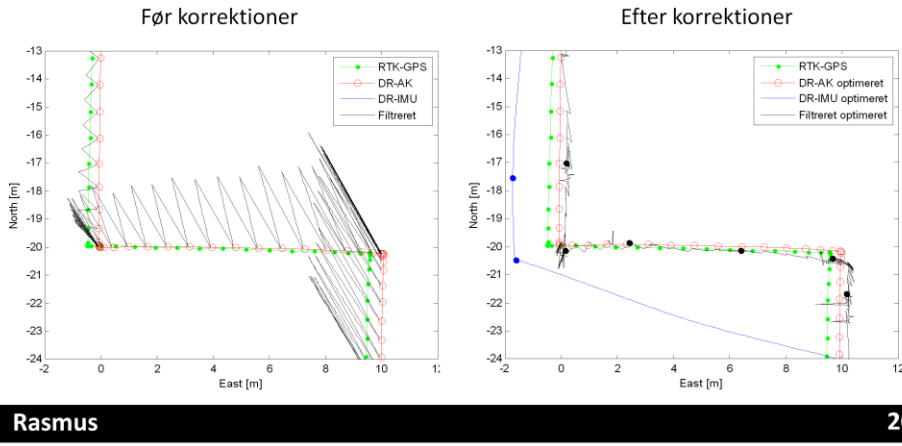
Man kan så stille spørgsmål ved, hvorfor resultaterne ikke forbliver gode?

Dette kan skyldes mange parametre, f.eks.

- at vi kun har korrigeret for den systematiske fejl, Bias
- at blæsten gav udslag på målingerne med inklinometeret, men at vi ikke tager højde for tilfældige fejl som elevation og bank kan være observeret med
- eller at vognen vibrerer under kørsel, og dermed ikke har præcis den hældning som vi bruger ved afsætning

## Resultat vedrørende filtreret positioner

- Synligt forbedret filtrerede positioner efter korrektioner foretaget
- Ses at DR-AK vægtes nu mindre
- Stadig plads til forbedringer, eks. vægtningen af DR-IMU



Rasmus

20

Hvordan har disse ændringer så forplantet sig i den position som Kalman filtreres?

### **Grafer:**

Figuren til venstre viser positionsbestemmelserne før korrektionerne, og svarer til det setup, der er benyttet i rapporten, mens højre viser positionsbestemmelserne med de beskrevne ændringer.

Det overordnede resultat er, at den filtrerede position nu ikke afviger nær så meget fra den sande rute.

### **Klik med musen**

Dette ses især af den sidste del af den tilbagelagte rute, hvor de filtrerede positioner hvor IMU'en kun indgår, nu ikke forsvinder i en bestemt retning, men faktisk følger den retning der tilbagelægges.

Det ses også at vægtningen af DR-AK ikke er nær så stor som før. Det skyldes at vi siger spredningen på IMU positioner reelt nulstilles når orienteringen er blevet opdateret

### **Højre graf:**

Der er dog stadig plads til forbedringer, for vægtningen af IMU'en er tilsyneladende ikke korrekt, da den filtrerede position nu generelt er dårligere end DR-AK alene, da afstanden til den "sande" rute nu er større til sidst.

## Disposition

- Velkomst og præsentation af projektet (Lars)
- Deviationstabel (Lars)
- Ny test med optimering af parametre (Rasmus)
- Bias – ustabilitet og betydning (Carsten)
- Forbedring af systemet (Carsten)

## Bias på IMU'en

- Bias er en systematisk fejl
  - Giver et output selvom input er nul
- Systemet bliver kalibreret for denne fejl
  - Automatisk indsamling af data til kalibrering
  - Den eneste systematiske fejl vi har kalibreret for
- Data til kalibrering er indsamlet inden hver testkørsel
  - Mange bud på bias
  - Mulighed for sammenligning

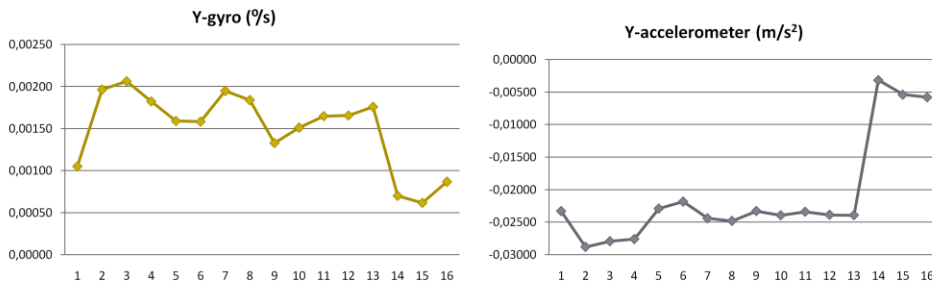
I kalibreringen af IMU'en, har vi udelukkende kalibreret for en af de systematiske fejl, nemlig bias. Denne fejl giver et konstant output, selvom inputtet er nul.

For at kunne kalibrere for bias, er der lavet en dataindsamling, der kører et minut, hvorefter der beregnes bias på baggrund af disse data.

Da vi har indsamlet data til kalibrering forud for hver testkørsel, har vi mange bud på værdier for bias, hvilket har muliggjort en sammenligning.

## Bias på IMU'en

- Ikke stabil over tid
  - Flere årsager
  - Temperaturskift
  - Vind
- Markant ændring ved ny test



Carsten

23

Figurene viser bias for hhv. gyroen og accelerometeret på Y-aksen. Det ses, at der er 16 forskellige værdier. De 13 første er indsamlet under de oprindelige testkørsler, hvor IMU'en ikke har været slukket mellem testkørslerne, mens de sidste tre er indsamlet under den nye test, ligeledes uden at IMU'en har været slukket.

Som vi kan se, er bias ikke stabil over tid, selvom IMU'en ikke har været slukket mellem dataindsamlingerne. Der kan være flere årsager til dette.

Den primære årsag er, at der er store temperaturændringer mellem indsamlingen af data til kalibreringen. Mellem de to første værdier er der et spring på ca. 10 grader, da den første er foretaget indenfor, mens den anden er foretaget udenfor.

Derudover kan vinden have påvirket dataindsamlingen udenfor.

Det ses tydeligt, at der er en markant ændring af værdierne for bias ved den nye test, hvilket understøtter teorien om, at værdierne for bias kan ændre sig fra opstart til opstart af IMU'en.

## Bias på IMU'en

- Bias har stor betydning for den endelige position
  - Beregning af forskellen i bias på kalibreringsdata, KloakInde1 og KloakInde2
  - Begge foretaget inde, ingen vind eller andre vibrationer
  - Temperaturforskel på 12 grader
  - Differens:
    - X-acc = -0,00009 (m/s<sup>2</sup>)
    - Y-acc = 0,00550 (m/s<sup>2</sup>)
    - Z-acc = -0,00361 (m/s<sup>2</sup>)
  - Giver en fejl på 0,003 meter efter et sekund
  - Fejl på 11,595 meter efter et minut

Det er altså tydeligt, at bias ændrer sig hele tiden, også over korte tidsrum. For at finde ud af hvor stor betydning en fejl i bias har for den endelige position, har vi foretaget en testberegning.

Vi har anvendt kalibreringsdata fra de to tests, KloakInde1 og KloakInde2. Begge disse dataindsamlinger er foretaget indenfor, så de burde være upåvirkede af vind og andre vibrationer og derfor give det bedste bud på bias på sensorerne.

IMU'en har ikke været slukket mellem de to dataindsamlingerne, men der har været et temperaturskift på 12 grader. Dette giver følgende differenser i værdierne for bias på accelerometrene. (Henvis til slide)

Herefter er der foretaget en positionsbestemmelse udelukkende med disse værdier.

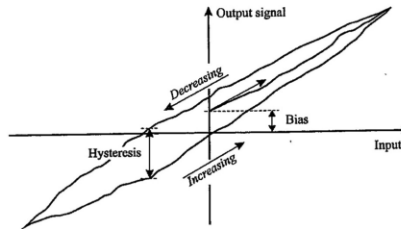
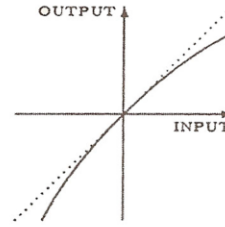
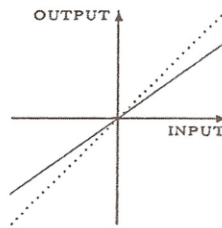
Dette giver en fejl på tre millimeter efter et sekund og en fejl på 11,5 meter efter bare et minut. I dette er der ikke taget højde for bias fejl på gyroerne, der ligeledes vil bidrage med fejl i positioneringen.

Dette understreger vigtigheden af, at have en korrekt bestemmelse af bias, inden der foretages en dataindsamling til positionsbestemmelse. For at undgå at værdierne for bias skifter hele tiden, vil det desuden være en god ide at lave temperaturstyring på IMU'en



## Andre fejl

- Systematiske fejl
  - Scale faktor
  - Nonlinearity
  - Hysteresis
- Tilfældige fejl
  - Random walk
  - White noise



Udover bias er der en række andre fejl der knytter sig til IMU'en. Af disse kan nævnes Scale factor, nonlinearity og hysteresis, der alle giver systematiske fejl på outputtet fra IMU'en

Derudover er der tilfældige fejl på IMU'en såsom Random walk, der betyder at der hele tiden registreres små ændringer i f.eks. Drejning og endelig white noise, der er meget almindeligt i elektroniske kredsløb.

Dette viser, at der stadig er plads til forbedringer i det udviklede system, da der ikke er kalibreret for andre systematiske fejl end bias.

## Disposition

- Velkomst og præsentation af projektet (Lars)
- Deviationstabel (Lars)
- Ny test med optimering af parametre (Rasmus)
- Bias – ustabilitet og betydning (Carsten)
- Forbedring af systemet (Carsten)

## Udvikling af systemet

- Det udviklede system er stort
  - Anvendt til at vise principperne
  - Kan ikke komme ned i en kloak
- For endelig anvendelse
  - Mindre system
  - Mindre instrumenter
  - Integration af andre sensorer
  - Udvidelse med kortmodul



Carsten

27

Udover forbedringer i kalibreringen, kan systemet også forbedres på hardware fronten.

Det system vi har arbejdet med er fysisk stort, og det kan selvsagt ikke komme ned i kloakker af den størrelse, som Arsløff ønsker. I vores tilfælde er den store vogn anvendt til at kontrollere og vise principperne inden for sensorintegrationen og positionsbestemmelsen.

I tilfælde af, at systemet skal tages i anvendelse, er det nødvendigt at videreudvikle det. Dette indebærer at lave et meget mindre system, hvor der anvendes mindre instrumenter, integreres andre sensorer og eventuelt udbygges softwaren med et kortmodul, så de beregnede positioner kan vises i et overordnet koordinatsystem. I udbygningen med andre og flere instrumenter, er det selvfølgelig nødvendigt, at tage hensyn til pris og størrelse inden der indkøbes nye instrumenter.

## IMU og Gyrokompas

- 3DM-GX1 (Microstrain)
  - Modifieret af Arsløff
  - Passer i kloakken
  - Indeholder både IMU og magnetometre
  - Påvirket af lokale magnetfelter
- Gyrokompas
  - Bliver ikke påvirket af magnetfelter
  - Primært udviklet til maritim brug
  - Ofte tunge og pladskrævende



Et muligt alternativ til den anvendte IMU, 3DM-GX1 fra firmaet Microstrain. Som vi kan se på billedet har Arsløff allerede haft gang i denne IMU og modificeret den, så den passer i størrelse på de mindste af deres inspektionstraktorer. Det særlige ved denne IMU er, at den også indeholder tre magnetometre, der er med til at stabilisere outputtet fra gyroerne.

Ulempen ved anvendelse af magnetometre er, at de stadig er påvirket af lokale magnetfelter, og det er derfor stadig nødvendigt med ekstra sensorer i systemet, for at opnå tilstrækkeligt nøjagtige positioner.

Et alternativ til et magnetisk kompas eller magnetometre, kan være et gyrokompas. På grund af deres opbygning og funktionsmetode, bliver de ikke påvirket af magnetfelter og de er derfor meget retningsstabile.

De er dog oftest udviklet til maritim brug, hvilke ofte gør dem tunge og pladskrævende.

Der er dog enkelte eksempler på gyrokompasser, der har en størrelse, der muliggør anvendelse i et system, der er lille nok, til at kunne gå ned i en kloak. F.eks....

## Orienteringssensor

- Orienterings sensor (Xsens)
- Indeholder:
  - Accelerometre
  - Gyroer
  - Magnetometre
- Kan ifølge Xsens levere
  - Real-time drift fri 3D orientering
  - Kalibreret 3D accelerationer
  - Kalibreret 3D vinkelændringer
  - Kalibreret 3D magnetfeltsdata



Derudover har firmaet Xsens udviklet hvad de kalder en orienteringssensor. Den er i realiteten stort set magen til 3DM-GX1 fra Microstrain, da som denne ligeledes indeholder tre accelerometre, tre gyroer og tre magnetometre. Det virker dog umiddelbart som om, at signalbehandlingen i dette instrument, er bedre end i 3DM'en.

Dette resulterer ifølge Xsens i, at der i Real-time kan fås en 3D orientering, der er fri for drift, men derudover fås der kalibrerede accelerationer, vinkelændringer og magnetfeltsdata, alt sammen i 3D. Med baggrund i disse data, vil det være muligt at foretage en positionsbestemmelse, nøjagtig som vi har gjort med vores system, men forhåbentlig med bedre resultater, hvis data virkelig er fri for drift.

Vi har set en præsentationsvideo af instrumentet, som viser meget lovende resultater i orienteringsbestemmelsen, men det må forventes, at magnetometrene i dette instrument fortsat har problemer med lokale magnetfelter.

# Inklinometre

- Inklinometre
  - Driftsikre
  - Robuste
  - Ikke påvirket af magnetfelter
  - Driver ikke over tid
  - Har en fornuftig størrelse
- H4PD1-22 (Rieker)
  - Dual-axis
  - 5 cm i diameter
  - $\pm 22^\circ$  med en nøjagtighed på  $0,1^\circ$
- CXTILT 02 (Crossbow)
  - Dual-axis
  - 10 cm bred
  - $\pm 75^\circ$  med en nøjagtighed på  $0,2^\circ$



Som nævnt i rapporten (og denne fremlæggelse), vil det give en forbedring af systemet, hvis der løbende kunne opdateres med bank og elevation. Til dette formål kan der anvendes inklinometre, der netop måler hældninger i forhold til vandret, altså bank og elevation. Inklinometre er ofte driftsikre og robuste og er ikke påvirket af magnetfelter, da de måler i forhold til Jordens tyngdekraft. Dermed driver de heller ikke over korte tidsrum og de har efterhånden en størrelse der gør, at de kan anvendes i vores system.

Der kan f.eks. fås et inklinometer fra firmaet Rieker, der måler både på X- og Y-aksen, er meget robust fordi det ikke har nogen bevægelige dele og samtidig kun har en diameter på fem cm. Det kan måle inden for et interval på  $\pm 22$  grader, men en nøjagtighed på  $0,1$  grad.

Eller et CXTILT 02 fra Crossbow, der ligeledes måler på X- og Y-aksen, heller ikke har nogen bevægelige dele, er lidt større end den anden, men den kan til gengæld måle inden for et interval på  $\pm 75$  grader med en nøjagtighed på  $0,2$  grader.

Dette er bare to eksempler på inklinometre, der kan anvendes til vores formål, og som også har en fornuftig størrelse.

## Perspektivering

- Mange muligheder for videreudvikling
- Software
  - Kortmodul
  - Databehandling
- Hardware
  - Nye instrumenter
  - Mindre
  - Mere nøjagtige

Selvom vi synes, at vi er nået et langt stykke af vejen mod en positioneringssystem, der kan anvendes i rørsystemer, er det ud fra denne præsentation tydeligt, at der stadig er mange muligheder for videreudvikling. Det gælder både softwaredelen, der kan udvides med kortmodul og en forbedret databehandling, så positioneringen foregår Real-time, så der anvendes en anden form for filtrering, eller der kan anvendes andre, og mere præcise bevægelsesmodeller. Men også hardware-delen kan udvikles med nye instrumenter, hvor der er mange instrumenter, der både er mindre og mere nøjagtige end dem vi har arbejdet med. Der er altså nok at gå i gang med, i en videreudvikling af dette system.

Slut

Slut

Spørgsmål...?

Gruppe 09ms1002

32

Det var alt vi havde at sige lige nu, medmindre der er nogle spørgsmål?