MEDICAL UNIVERSITY OF VIENNA

TECHNISCHE UNIVERSITÄT WIEN Vienna | Austria

# THREE DIMENSIONAL MOTION TRACKING FOR ASSESSMENT OF MOTOR TASKS

Master Thesis

for obtaining the academic degree

**Master of Science**

in

**Biomedical Engineering**

submitted by

**Thomas Bachner**
0926726

at the faculty of "Mechanical and Industrial Engineering" at the Vienna University of Technology

Supervision:
Professor Ph.D. Winfried Mayr
Ph.D. Matthias Krenn

Vienna, JJJJ.MM.TT

(signature author) (signature supervisor)

# STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..

date (signature)

# Abstract

The Microsoft Kinect$^{TM}$ sensor is a gaming device, developed for a control-free gaming experience, using motion tracking. When this technology was released, other fields of application got interested in this technology. Therefore, a lot of papers have been published, in which the capabilities of the Kinect$^{TM}$ sensor were studied. Based on these results, the Microsoft Kinect$^{TM}$ was used for medical applications. One of them was rehabilitation, where the sensor was used for gesture detection and for checking the performed exercises.

This thesis describes the issue, whether the Microsoft Kinect$^{TM}$ sensor can be used for a more detailed human motion analysis, such as calculating the flexion of the elbow. Therefore, the functions and commands provide from the Kinect$^{TM}$ sensor are inspected and investigated, in order to create a program, which delivers the desired or calculated parameters. In addition, obstacles and limitations, deriving during the development of the program are explained.

By performing different measurements, namely gait analysis and hand motion & grasping tasks (focusing on hand motion & grasping), the accuracy and reliability of the obtained data can be shown. As comparison, the data of an electrogoniometer are given, to determine the limitations and restrictions of the Microsoft Kinect$^{TM}$ sensor, compared to a state of the art method for human motion analysis.

Der Microsoft Kinect™ Sensor wurde für die Konsole X-Box entwickelt und soll eine Spielerfahrung ohne Kontroller mittels Bewegungserkennung ermöglichen. Nach seiner Veröffentlichung wurde diese neue Art von Technologie auch für andere Anwendungsgebiete interessant. Zahlreiche Papers wurden publiziert, in welche die Fähigkeiten bzw. Performance des Kinect™ Sensors untersucht wurden. Basierend auf diesen Ergebnissen wurde auch der Einsatz des Sensors im medizinischen Bereich diskutiert. Einer dieser Bereiche ist die Rehabilitation, in der, der Sensor zur Gestenerkennung bzw. zur Überprüfung von Übungen verwendet wurde.

Diese Arbeit befasst sich mit der Frage, ob der Microsoft Kinect™ Sensor für genauere Analysen der menschlichen Bewegung, wie zum Beispiel der Berechnung der Flexion des Ellbogens, eingesetzt werden kann. Aus diesem Grund werden die Funktionen und Befehle des Sensors untersucht, um darauf aufbauend, ein Programm zu entwickeln, welches die gewünschten Parameter berechnet bzw. ausliest. Zusätzlich wird noch auf etwaige Hindernisse, die während der Entwicklung des Programmes auftreten, erläutert.

Durch die Durchführung unterschiedlicher Messungen, wie Ganganalyse, Handbewegungen & Greifaufgaben (wobei der Schwerpunkt auf Handbewegungen & Greifaufgaben liegt), lässt sich die Genauigkeit und Zuverlässigkeit der Daten untersuchen. Zur Bestimmung dieser Eigenschaften bzw. der Limitierungen des Microsoft Kinect™ Sensors werden diese Daten mit den Daten eines Elektrogoniometers verglichen.

# Content

# Introduction

Human motion analysis is the tracking of motor tasks with cameras. More precisely, it is the detection of human activities and actions to insure e.g. the safety of people in public places, to monitor people working in a hazardous environment or to check the conditions of the folks in a “smart” home environment (C. Sankaranarayanan, Chellappa, & Baraniuk, 2011).

There are several systems with different methods available on the market, which provide sufficient accuracy & reliability for their field of work (Wang, Hu, & Tan, 2003). However, these systems are very expensive and therefore cannot be tested out of curiosity.

The Microsoft Kinect™ sensor is a low cost system for human motion analysis, used commercially for video games (Zhang, 2012). A lot of research was done with this sensor in order to investigate if it can be used for other topics and how is the accuracy & reliability compared to the state of the art systems (Weber et al., 2012) (Wei, Zhang, & Chai, 2012). One of this fields is medicine, where the Microsoft Kinect™ sensor was used for different rehabilitation issues (Mousavi Hondori & Khademi, 2014), such as home-based rehabilitation (Su, 2013).

Most of these studies showed, that the Microsoft Kinect™ sensor has a lot of potential, but was not able to compete with the current systems at the moment, however, they were confident, that further developments of the Kinect™ would have a great impact in medical applications (Weber et al., 2012) (Wei et al., 2012).

In 2013, Microsoft released an improved version of the Microsoft Kinect™ sensor (Zennaro et al., 2013). Tests between the older and the newest version of the Kinect™ sensor showed, that the newest version is overall better (e.g. area of detection, rotation accuracy) as its predecessor (Amon & Fuhrmann, 2014) (Zennaro et al., 2013).

This thesis is concerned with the issue whether this new generation is capable to record simple motor tasks for medical applications and how accurate & reliable it is compared to conventional devices. Therefore, a program will be developed, which provides the sufficient functions for three dimensional motion tracking. Later this program is used in different clinical measurements to examine the limitations, the accuracy & reliability of the Kinect™ sensor and compares it to conventional measurement methods.

# Literature Review

## Human Motion Analysis

The motion analysis is a relative interesting topic among the researchers of computer vision. It got popular in the 1960s (C. Sankaranarayanan et al., 2011) and can be divided into different subareas. One of this subareas is the so called human motion analysis.

This area describes the detection, tracking and recognition of people using cameras and computers (Wang et al., 2003). More precisely, there are different approaches for different applications in human motion analysis (Aggarwal & Cai, 1997):

1) Use of human body parts for motion analysis
2) Single view or multiple camera perspective to track human motion
3) Perceive human activities based on sequences

In this thesis the further descriptions put the focus on "Use of human body parts for motion analysis", because it is most important for medical purpose, such as marker based tracking and model based tracking.

## Marker Based Tracking

In a marker based system, markers are attached to the body parts of the participant and tracked by the surrounding infrared cameras. Such a system needs at least two cameras in order to recognize the markers in a three dimensional space. Additionally, the markers have to be calibrate for each session and synchronized to their respective cameras (Brown, Smallwood, Barber, Lawford, & Hose, 1999).

The huge advantage of a marker based system compared to other tracking systems is the high accuracy which can be achieved. In order to get this high accuracy, the markers have to be placed precisely on the body parts. Therefore, it is a really time consuming process (Ceseracciu, Sawacha, & Cobelli, 2014).

Another distinction is made between passive and active markers. Both methods are used uniformly in practical terms (Begg & Palaniswami, 2006).

### Active Marker Systems

Active markers have built in Light Emitting Diodes (LEDs), which emit visible or infrared light. This light will be detected by the surrounding cameras (Stathopoulou & Tsihrintzis, 2010). The LEDs are linked together and flash in a predetermined sequence. Because of that, the system is able to identify each single marker automatically. This is a great benefit compared to passive markers. However, the active markers need a power supply and they have to be synchronized with each other (Begg & Palaniswami, 2006).

### Passive Marker Systems

The camera lenses in a passive marker system are enclosed with infrared light illuminators. Light will be reflected on the surface of the markers and detected by the cameras. Therefore, each ray from the marker to the camera has to be identified and allocated. This process is very difficult, for which reason at least six cameras in a passive marker system are needed (Begg & Palaniswami, 2006).

## Model Based Tracking

Nowadays, many researchers are interested in model based tracking systems, also called visual based tracking systems. This method doesn't need any markers or other helping tools to detect people, it just needs a camera (Gavrila, 1999).

A captured sequence of images is used to estimate a human body model, consisting of segments connected by joints and the shape of the human (Figure 1).

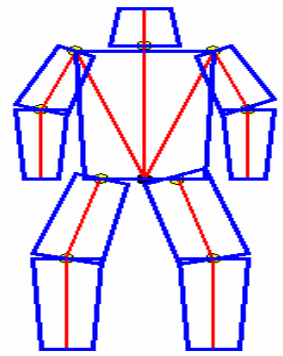


Figure 1: Example of a human body model (reprinted from (Huang & Huang, 2002))

Further, image descriptors are extracted from the original visual input to represent the appearance of the human, such as edges, colors or silhouettes (Poppe, 2007). Because model based tracking systems don't need any markers and large numbers of cameras the system is relatively cheap. Also, there are quasi no preparations compared to marker based systems (Ceseracciu et al., 2014). However, the accuracy of the system is rather low (Ceseracciu et al., 2014) and by using only a single camera depth ambiguities can appear (Poppe, 2007).

## Microsoft Kinect™ Sensor

The Microsoft Kinect™ sensor was designed for the Xbox 360 as a control loose entertainment system and got published in November 2010 (Zhang, 2012). The great potential which derives from this system was early recognized. Therefore, other scientific areas, such as healthcare applications got interested (Zhang, 2012). As a result, Microsoft developed a "Kinect for Windows Developer Toolkit", allowing the user to manipulate and to communicate with the Microsoft Kinect™ sensor via PC (Soares Beleboni, 2014).

One of the medical topics that could benefit by using the Kinect™ is for example rehabilitation. Because of that, a lot of papers about rehabilitation systems using the Microsoft Kinect™ sensor and checking the accuracy & reliability of these systems were published (Mousavi Hondori & Khademi, 2014).

In 2013 the second version of the sensor was released, providing a better resolution, a revised toolkit, and much more (Zennaro et al., 2013). A brief description about the differences is shown in Figure 2.

| **Feature** | **Kinect for Windows 1** | **Kinect for Windows 2** |
| --- | --- | --- |
| Color Camera | 640 x 480 @ 30fps | 1920 x 1080 @ 30fps |
| Depth Camera | 320 x 240 | 512 x 424 |
| Max Depth Distance | ~4.5 M | 8 M |
| Min Depth Distance | 40 cm in near mode | 50 cm |
| Depth Horizontal Field of View | 57 degrees | 70 degrees |
| Depth Vertical Field | 43 degrees | 60 degrees |
| Tilt Motor | yes | no |
| Skeleton Joints Defined | 20 joints | 25 joints |
| Full Skeletons Tracked | 2 | 6 |
| USB Standard | 2.0 | 3.0 |
| Supported OS | Win 7, Win 8 | Win 8 |
| Price | $249 | $199 |

Figure 2: Comparison of the Microsoft Kinect™ sensors (adapted from (Ashley, 2014))

The Microsoft Kinect™ sensor is a low cost marker less tracking system, which is capable for full-body 3D motion capturing (Soares Beleboni, 2014). It includes a depth sensor consists of an infrared projector and an infrared camera, and a color camera (RGB-camera) (Soares Beleboni, 2014). The front view of an opened Microsoft Kinect™ sensor, is shown in Figure 3.

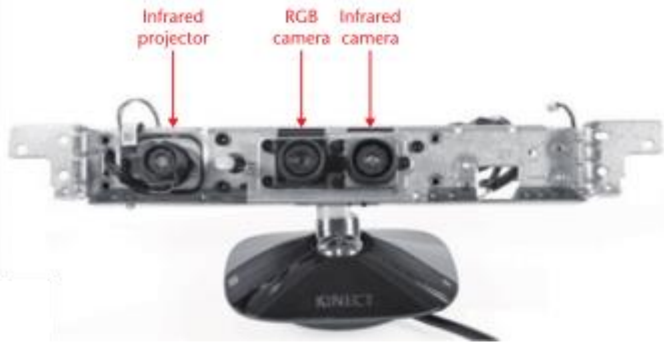



Figure 3: Components of Microsoft Kinect$^{TM}$ sensor v1 (reprinted from (Zhang, 2012))

The camera is able to track 25 different joints (Figure 4) from six persons at the same time, providing data such as three dimensional coordinates of the joints, the orientation of the joints and information about the hand state ((Microsoft, 2016b)). Furthermore, it can be used for real time facial expression or for face capturing (Alabbasi, Moldoveanu, & Moldoveanu, 2015).

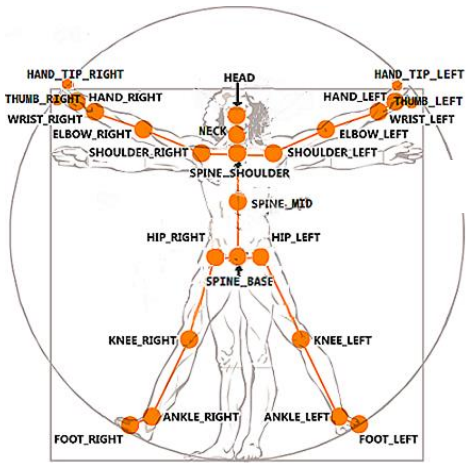



Figure 4: Trackable joins from the Microsoft Kinect$^{TM}$ sensor v2 (reprinted from (Microsoft, 2016b))

# Implementation

This chapter contains the implementation and processing for the program to detect simple motor tasks. It is discussed in more detail which data the Microsoft Kinect™ sensor provides, how they can be accessed and how these data can further be used. The manipulation of the sensor is done with the "Kinect for Windows Developer Toolkit" for the program "Visual Studio" and the programming language C#.

The parameters of interest that should be recorded and saved are:

- The positions of the joints
- The orientation angles of the joints
- The angle of the knee
- The angle of the hip
- The angle of the elbow

Additionally, the human based model and the color image of the tracked person is calculated and saved in a video file.

## Initialization of the Microsoft Kinect™ Sensor

Before the data of the Microsoft Kinect™ sensor can be used within a program, a few steps have to be done. First of all, the "Kinect for Windows Developer Toolkit" or other open source toolkit should be installed on the computer in order to provide the sufficient methods to communicate with the sensor. Hereafter, these methods are embedded into the program to get the data. The code for that process is shown below.

```
public void Initialization ()
{

  //Gets the Kinect sensor
  KinectSensor kinect = KinectSensor.GetDefault();

  //Opens the kinect sensor
  kinect.Open();

  //Multi source frame reader which catches different frames from the kinect
  MultiSourceFrameReader framereader =
        kinect.OpenMultiSourceFrameReader(FrameSourceTypes.Color | FrameSourceTypes.Body);

  //Event that fires when a frame is captured
  framereader.MultiSourceFrameArrived += Reader_MultiSourceFrameArrived;

}

//Called methode when the event fires
public void Reader_MultiSourceFrameArrived(object sender, MultiSourceFrameArrivdEventArgs e)
{

  //Do Something

}
```

To initialize the Microsoft Kinect™ sensor within the program the command `KinectSensor.GetDefault()` has to be used. Further, the frame reader of the corresponding sensor is defined in order to catch the frames coming from the Kinect™ and make them accessible for the user. For simple motor task tracking, the color frame and the body frame is captured, the color frame derives from the color camera, while the body frame contains the data calculated from the tracked body, such as joint positions, joint orientations, etc.

Each time the frame reader catches new frames from the specific sources, a method is called `Reader_MultiSourceFrameArrived()`. Within this method the user can manipulate the obtained data (e.g. visually represent the color frame or represent the joint positions of the tracked body). This process is done 30 times per second (30 Frames per second), therefore the manipulation of the acquired data should be completed within 33ms (1/30), otherwise there would be a loss of data. In order to ensure that, the parameters listed above are executed parallel by multiple threads.

# Recording & Calculating the Parameters of Interest

During a motion task session different parameters of the tracked person should be calculated and saved. This subchapter describes in detail, for each single parameter, how to get the corresponding data from the frame reader of the Kinect™ and how these data are manipulated to achieve the desired parameters.

## Displaying the Color Image and the Body Model

The color camera of the Kinect™ generates the color frames which are captured by the frame reader. In order to get the color frame out of the frame reader the method `ColorFrameReference.AcquireFrame()` is used. Afterwards, the acquired color frame with a resolution of 1920x1080 pixels is converted into an image source, with the same resolution and displayed on the monitor.

```
//Called methode when the event fires
public void Reader_MultiSourceFrameArrived(object sender, MultiSourceFrameArrivedEventArgs e)
{

  //Get the frames from the event
  var reference = e.FrameReference.AcquireFrame();

  Get color frame
  using (var frame = reference.ColorFrameReference.AcquireFrame())
        {

          //Convert color frame into ImageSource

          //Display ImageSource on the monitor

        }
}
```

The human body model is constructed from the joint positions stored in the body frames. The method to extract the body frames from the frame reader is similar to the color frames, `BodyFrameReference.AcquireFrame()`. The obtained three dimensional coordinates also called „camera space points“ are drawn as dots on a canvas and connected via lines, which is placed on top of the color image, representing the joints and bones.

The problem occurring by just placing the „camera space points“ on the color image is a shift between them (Figure 5). This is caused due to the spatial distance between the color camera and the infrared sensor.

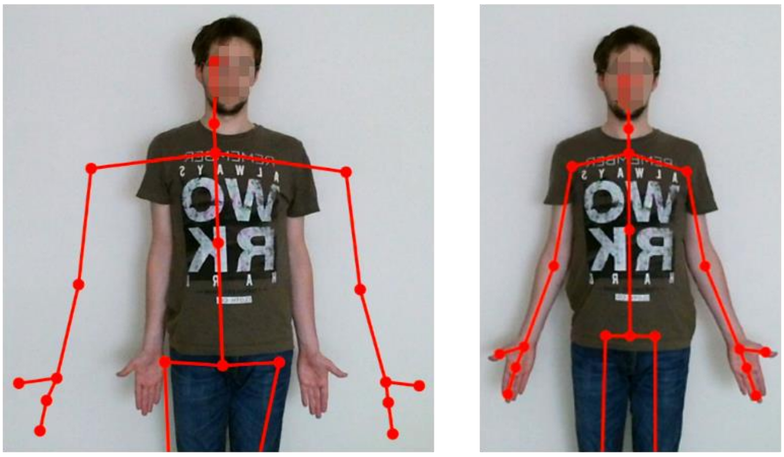


Figure 5: Image without and with using the coordinate mapper

In order to solve this problem Microsoft introduced in the “Kinect for Windows Developer Toolkit” a „coordinate mapper“, that converts the „camera space points“ into „color space points“ corresponding to the coordinates of the color image. This has to be done manually.

The code below shows the process how to acquire the coordinates from the body frame and how to converted them into „color space points“. To get a better impression it is only shown for one joint.

```
//Called methode when the event fires
public void Reader_MultiSourceFrameArrived(object sender, MultiSourceFrameArrivedEventArgs e)
{

  //Get the frames from the event
  var reference = e.FrameReference.AcquireFrame();

  //Get body frame
  using (var frame = reference.BodyFrameReference.AcquireFrame())
  {

    //Get the tracked bodies from the body frame
    IList<Body> bodies = new Body[frame.BodyFrameSource.BodyCount];

    //For each tracked body draw the joints
    foreach (var body in bodies)
    {

      if (body.IsTracked)
      {

       //Get the head from the tracked body
       Joint head = body.Joints[JointType.Head];

       //Convert the 3D camera space points into 2D color space points
       ColorSpacePoint colorPoint =
                kinect.CoordinateMapper.MapCameraPointToColorSpace(head.Position);

       //Draw the joint as dot at position colorPoint (X,Y) on the canvas

       //Draw the bone as line between the head and the predecessor (X,Y) on the canvas

      }
    }
  }
}
```

Additionally, to the human body model the hand state is drawn on the canvas as well, which are indicated by a colored circle within the hand, whereby the color depends on the state. The Microsoft Kinect™ sensor is able to distinguish between three different types of states by default - “open hand”, “closed hand” and “lasso”. In this program, the chosen colors of the states are green, red and blue. Figure 6 shows the illustration of the different hand states.


Figure 6: Different hand states

An important issue for the detection of the hand state is the positioning, the best results derive when the forehand is facing the camera, as it can be seen in Figure 6. Problems occur with the backhand, the side of the hand, or when the hand is partially covered. In these cases, the detection of the state is not always possible.

The hand state can be obtained from the body frames with the commands `body.HandRightState` and `body.HandLeftState` for the right/left hand and provides one of five possible states ("Unknown", "NotTracked", "Open", "Closed" & "Lasso"), which are used within a switch-case-branch. When the state is "Unknown" or "NotTracked" no circle is drawn on the canvas, while in the other three cases a different color is used for the circle. After that the circle is placed on top of the corresponding hand, using the position of the hands obtained with the command `Joints[JointType.HandLeft].Position` and `Joints[JointType.HandRight].Position` as the origin (Note: The "camera space points" have to be converted into "color space points" with the "coordinate mapper").

The implementation of the code is shown below, using only the right hand for a better view.

```
//For each tracked body draw right hand state
foreach (var body in bodies)
{

  if (body.IsTracked)
  {
    //Get the coordinates of the right hand
    CameraSpacePoint handRight = body.Joints[JointType.HandRight].Position;

    //Convert the 3D camera space points into 2D color space points
    ColorSpacePoint colorPoint =
             kinect.CoordinateMapper.MapCameraPointToColorSpace(handRight);

    //Get the state of the right hand
    switch (body.HandRightState)
    {
      case HandState.Open:

        //Draw green circle at position colorpoint

      break;

      case HandState.Closed:

        //Draw red circle at position colorpoint

      break;

      case HandState.Lasso:

        //Draw blue circle at position colorpoint

      break;
    }
  }
}
```

### Positions of the Joints in a Three Dimensional Space

These are the same "camera space points" which are used to create the human body model acquired from the body frames. During a motion session the data should be stored on the hard disk to make them accessible in the future. It must be noted, that the coordinates need to be arranged into a readable format and stored into a file (e.g. CSV file). The three coordinates of the joints can be accessed via the call `Joints[JointType].Position.X`, `Joints[JointType].Position.Y`, `Joints[JointType]..Position.Z`, where "JointType" is a placeholder for the different joint types tracked from the Microsoft Kinect™ sensor. The values of the coordinates are given in meters with the origin located at the center of the infrared sensor (Figure 7).


Figure 7: The camera space coordinate system (adapted from (Microsoft, 2016a))

The code below shows how the coordinates can be achieved. For a better understanding it is only shown for one joint.

```
public void Reader_MultiSourceFrameArrived(object sender, MultiSourceFrameArrivedEventArgs e)
{

  //Get the frames from the event
  var reference = e.FrameReference.AcquireFrame();

  //Get body frame
  using (var frame = reference.BodyFrameReference.AcquireFrame())
  {

    //Get the tracked bodies from the body frame
    IList<Body> bodies = new Body[frame.BodyFrameSource.BodyCount];

    //For each tracked body get the positions of the joints
    foreach (var body in bodies)
    {

      if (body.IsTracked)
      {

        float X = body.Joints[JointType.Head].Position.X;
        float Y = body.Joints[JointType.Head].Position.Y;
        float Z = body.Joints[JointType.Head].Position.Z;

        //Arrange X, Y and Z in a desired way

        //Write the positions into a file (e.g. csv)

      }
    }
  }
}
```

### Orientation Angles of the Joints in a Three Dimensional Space

The orientations of the different joints can be obtained from the body frames. However, these orientations consist of a four dimensional quaternion and not as desired of three angles in x-, y- and z-direction. In order to get the angles from the quaternion some calculations have to be done.

The quaternion can be accessed with the command `JointOrientation[JointType].Orientation`, where “JointType” is a placeholder for the different joint types tracked from the Microsoft Kinect™ sensor. Quaternions are representing angles of rotations, which are used to rotate a vector at a certain amount.

Next, three vectors in x-, y- and z-direction are generated, rotated by the quaternion and translated to the corresponding body joint, describing the orientation of this joint. The coordinates of the resulting orientation vectors are given as “camera space points”, thus they have to be converted into “depth space points” via the “coordinate mapper” for subsequent actions.

To calculate the orientation angles for the given body joint, the orientation vectors of the parent joint are required (e.g. child: elbow, parent: shoulder), using the same steps discussed above, leading to two joints with different vectors for the x-, y- and z-direction. The angle between the vectors for the same direction (e.g. vector in x-direction for the elbow and the shoulder) gives the orientation angle of the child joint in this direction in degree [°] using the formula

$$\theta = atan2\left(\frac{\Delta y}{\Delta x}\right) * \left(\frac{180}{\pi}\right)$$

where Δx and Δy are the difference of the vectors in a 2D coordinate system (Note: x and y are the coordinates of the vectors). The term $\frac{180}{\pi}$ changes the unit from radiant [rad] to degree [°].

For a better understanding for the calculation of the orientation angles Figure 8 shows the process in a graphical view.

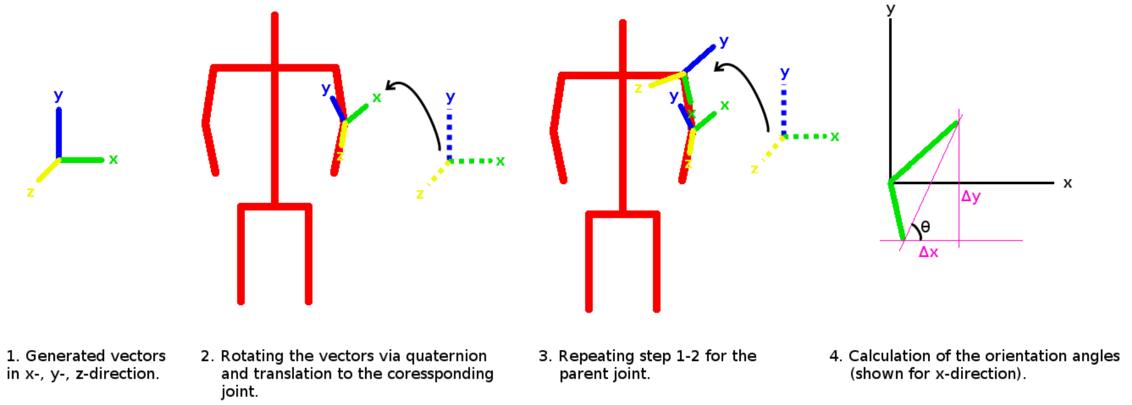



Figure 8: Calculation of the orientation angles for the left elbow

## Calculation of the Knee Angle

The knee connects the thigh with the lower leg. It is a hinge joint, which can only be stretched and bended, thus it has one degree of freedom. The bending and stretching of the knee is characterized via the knee angle, which can mathematically be described as the angle between two 3D vectors (representing the thigh and the lower leg) and calculated by the arccosine of the dot product of these two vectors.

Unfortunately, the Kinect™ sensor doesn't provide any information on the limbs themselves, only on the joints. Therefore, the 3D coordinates ("camera space points") of the joints are used to calculate the vectors for the thigh and the lower leg. A simplified overview for all steps is given in Figure 9.

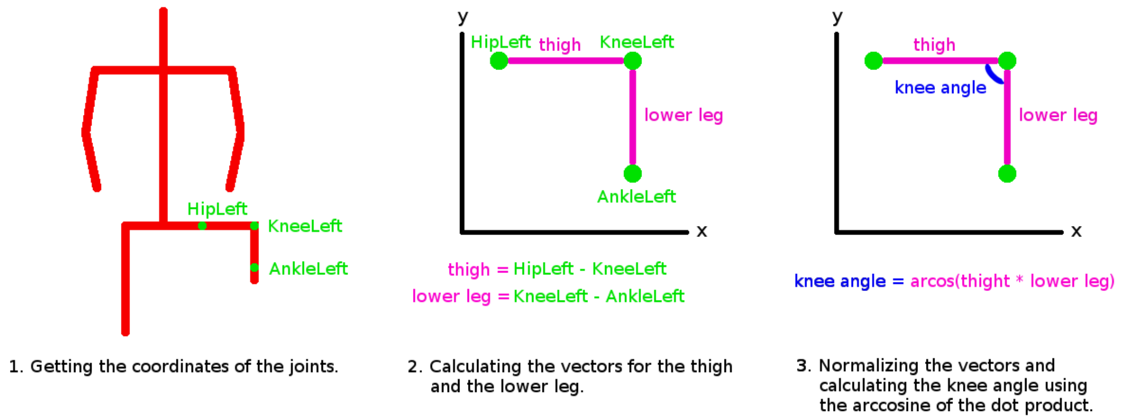



Figure 9: Calculation of the left knee angle

The joint coordinates used for the calculation derives from the hip, knee and ankle obtained from the body frames. They can be accessed with the command `Joints[JointType].Position`, where “JointType” is a placeholder for “HipLeft”, “KneeLeft”, “AnkleLeft”, “HipRight”, “KneeRight” & “AnkleRight”. The vector of the thigh can be calculated by subtracting the knee from the hip, and the vector of the lower leg is the subtraction of the ankle from the knee. The knee angle is the arccosine of the dot product of these two vectors. Additionally, the calculated data are saved in a CSV file, containing a timestamp and the angles.

The code below shows the computation of the knee angle, for a better view it is only shown for the left one.

```
//For each tracked body calculate the knee angles
foreach (var body in bodies)
{
  if (body.IsTracked)
  {

  //Get the coordinates of the hip, knee & ankle
  CameraSpacePoint hipLeft = body.Joints[JointType.HipLeft].Position;
  CameraSpacePoint kneeLeft = body.Joints[JointType.KneeLeft].Position;
  CameraSpacePoint ankleLeft = body.Joints[JointType.AnkleLeft].Position;

  //Convert the obtained coordinates into 3D vectors
  Vector3D hipLeftVect = new Vector3D(hipLeft.X, hipLeft.Y, hipLeft.Z);
  Vector3D kneeLeftVect = new Vector3D(kneeLeft.X, kneeLeft.Y, kneeLeft.Z);
  Vector3D ankleLeftVect = new Vector3D(ankleLeft.X, ankleLeft.Y, ankleLeft.Z);

  //Calculate the thigh and the lower leg
  Vector3D thighVect = hipLeftVect - kneeLeftVect;
  Vector3D lowerLegVect = kneeLeftVect - ankleLeftVect;

  //Calculate the knee angle
  thighVect.Normalize();
  lowerLegVect.Normalize();

  double dotProduct = Vector3D.DotProduct(thighVect, lowerLegVect);

  double leftKneeAngle = Math.Acos(dotProduct) / Math.PI * 180;

  }
}
```

### Calculation of the Hip Angle

The range of motion of the hip is greater than the one of the knee consisting of flexion & extension, abduction & adduction and rotation, therefore it has three degrees of freedom. Further, the hip is a so called ball joint connecting the pelvis and the thigh.

The hip angle is defined as the angle between the body and the leg, regardless of whether it is flexion & extension or abduction & adduction. Of course more information could be acquired by differentiating between these two movements, but it is enough to know the overall angle compared to the initial position. Under these assumptions, the hip angle can mathematically be described similar to the knee angle, as the arccosine of the dot product of two 3D vectors (representing the thigh and the body).

The Microsoft Kinect™ sensor provides the 3D coordinates (“camera space points”) of the hip and the knee to calculate the vector for the thigh, as well as the 3D coordinates (“camera space points”) of the spine base and the spine mid for the body vector. These two vectors have no point in common, therefore the end point of one of these vectors has to be translated onto the start point of the other one. With this approach the computation of the hip angle is independent from the position of the tracked person in front of the Kinect™ sensor. A simplified overview for all steps is given in Figure 10.

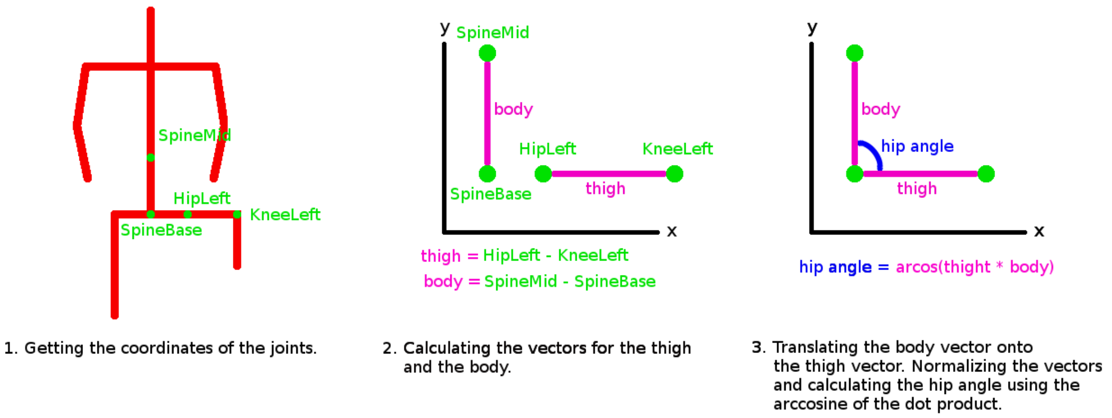



Figure 10: Calculation of the left hip angle

The joint coordinates used for the calculation derives from the hip, knee, spine mid and spine base obtained from the body frames. They can be accessed with the command `Joints[JointType].Position`, where “JointType” is a placeholder for “HipLeft”, “KneeLeft”, “HipRight”, “KneeRight”, “SpineMid” & ”SpineBase”. The vector of the thigh can be calculated by subtracting the knee from the hip, and the vector of the body is the subtraction of the spine base from the spine mid. Afterwards the body vector is translated onto the thigh vector, thereby the hip angle can be represented by the arccosine of the dot product of these two vectors. Additionally, the calculated data are saved in a CSV file, containing a timestamp and the angles.

The code below shows the computation of the hip angle. For a better view it is only shown for the left one.

```
//For each tracked body calculate the hip angles
foreach (var body in bodies)
{
  if (body.IsTracked)
  {

    //Get the coordinates of the hip, knee, spine mid & spine base
    CameraSpacePoint hipLeft = body.Joints[JointType.HipLeft].Position;
    CameraSpacePoint kneeLeft = body.Joints[JointType.KneeLeft].Position;
    CameraSpacePoint spineMid = body.Joints[JointType.SpineMid].Position;
    CameraSpacePoint spineBase = body.Joints[JointType.SpineBase].Position;

    //Convert the obtained coordinates into 3D vectors
    Vector3D hipLeftVect = new Vector3D(hipLeft.X, hipLeft.Y, hipLeft.Z);
    Vector3D kneeLeftVect = new Vector3D(kneeLeft.X, kneeLeft.Y, kneeLeft.Z);
    Vector3D spineMidVect = new Vector3D(spineMid.X, spineMid.Y, spineMid.Z);
    Vector3D spineBaseVect = new Vector3D(spineBase.X, spineBase.Y, spineBase.Z);

    //Calculate the thigh and the body
    Vector3D thighVect = hipLeftVect - kneeLeftVect;
    Vector3D bodyVect = spineMidVect - spineBaseVect;

    //Translation of the body vector
    Vector3D bodyVectTrans = bodyVect + hipLeftVect;

    //Calculate the hip angle
    thighVect.Normalize();
    bodyVectTrans.Normalize();

    double dotProduct = Vector3D.DotProduct(bodyVectTrans, thighVect);

    double hipAngle = Math.Acos(dotProduct) / Math.PI * 180;

  }
}
```

### Calculation of the Elbow Angle

The joint of the elbow consists of three sub joints, allowing flexion/extension and pronation/supination. It connects the upper arm with the forearm and has two degrees of freedom. The elbow angle is defined as the angle between the upper arm and the forearm caused by flexion/extension. Therefore, it can mathematically be described as the angle between two 3D vectors (representing the upper arm and the forearm) and calculated by the arccosine of the dot product of these two vectors, the same way it was done for the knee and the hip. A simple overview of the calculation can be found in Figure 11.

The pronation/supination between ulna and radius is described within the orientation angles and is visually displayed on the interface of the program.

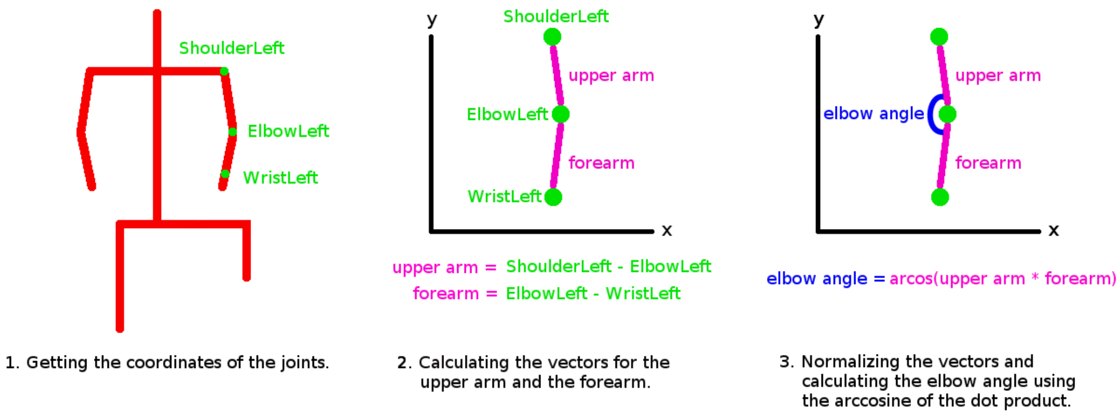



Figure 11: Calculation of the left elbow angle

From the body frames the joint coordinates of the shoulder, the elbow and the wrist can be obtained. To access these data the command `Joints[JointType].Position`, where "JointType" is a placeholder for "ShoulderLeft", "ElbowLeft", "WristLeft", "ShoulderRight", "ElbowRight" & "WristRight", is used. By subtracting the elbow from the shoulder, the vector of the upper arm is calculated, while the subtraction of the wrist from the elbow leads to the vector of the forearm. The elbow angle is the arccosine of the dot product of these two vectors. Additionally, the calculated data are saved in a CSV file, containing a timestamp and the angles.

The code below shows the computation of the elbow angle, for a better view it is only shown for the left one.

```
//For each tracked body calculate the elbow angles
foreach (var body in bodies)
{
  if (body.IsTracked)
  {

  //Get the coordinates of the shoulder, elbow & wrist
  CameraSpacePoint shoulderLeft = body.Joints[JointType.ShoulderLeft].Position;
  CameraSpacePoint elbowLeft = body.Joints[JointType.ElbowLeft].Position;
  CameraSpacePoint wristLeft = body.Joints[JointType.WristLeft].Position;

  //Convert the obtained coordinates into 3D vectors
  Vector3D shoulderLeftVect = new Vector3D(shoulderLeft.X, shoulderLeft.Y, shoulderLeft.Z);
  Vector3D elbowLeftVect = new Vector3D(elbowLeft.X, elbowLeft.Y, elbowLeft.Z);
  Vector3D wristLeftVect = new Vector3D(wristLeft.X, wristLeft.Y, wristLeft.Z);

  //Calculate the upper arm and the forearm
  Vector3D upperArmVect = shoulderLeftVect - elbowLeftVect;
  Vector3D forearmVect = elbowLeftVect - wristLeftVect;

  //Calculate the elbow angle
  upperArmVect.Normalize();
  forearmVect.Normalize();

  double dotProduct = Vector3D.DotProduct(upperArmVect, forearmVect);

  double leftElbowAngle = Math.Acos(dotProduct) / Math.PI * 180;

  }
}
```

### Saving the video and the data

The described parameters (joint positions, joint orientation angles, hip angles, knee angles & elbow angles) as well as the visual output (color frames, human body model, hand states & joint orientations) derived from a recording session, should be saved on the hard disk for subsequent analysis and evaluation. It should also be mentioned, that the images are vertically mirrored due to the Kinect™ sensor.

The method provided from Microsoft for the Kinect™ sensor uses the “Kinect Studio” API to store the raw data (e.g. color frames, body frames) in a single file as XEF format. This file works like a virtual Microsoft Kinect™ sensor, that can be loaded with the “Kinect Studio” or any program that uses the API.

Problems occurring with this method:

- Huge file size (several gigabytes for a few seconds).
- Files cannot be opened with a normal video player, because it is not a video.

- The desired parameters (joint positions, hip angles, etc.) must be recalculated from the raw data (calculation time depends on the duration of the recording session).

This method is insufficient, because it doesn't save single parameters or videos. They have to be extracted from the raw data in an additional step, that could also be done directly during a recording session without creating the XEF file. Therefore, the program saves the parameters and the videos directly on the hard disk as follows:

To generate the videos, the Microsoft Expression™ Encoder 4 API is used within the program to capture the screen. Thereby, the color image, the human body model, the joint orientations and the hand states are recorded. Furthermore, the program should be displayed on the monitor the complete recording time, otherwise the different actions are visible in the video afterwards. The file is saved within the output folder as a XESC file, that has to be converted manually into a common video format (e.g. WMV) using the Microsoft Expression™ Encoder 4 software.

The corresponding commands to initialize, to start and to stop the screen capture are shown in the code:

```
//Screen capture object
private ScreenCaptureJob job;

//Initialize the screen capture
public void ScreenCaptureInitialization()
{

  job = new ScreenCaptureJob();

  //Define and set the range of the screen that should be captured
  System.Drawing.Size monitorSize = SystemInformation.PrimaryMonitorSize;

  System.Drawing.Rectangle capRect =
                new System.Drawing.Rectangle(0, 0, monitorSize.Width, monitorSize.Height);

  job.CaptureRectangle = capRect;

  //Set the save folder
  job.OutputPath = "Output";

}

//Start the screen capture
public void ScreenCaptureStart()
{
  job.Start();
}

//Stop the screen capture
public void ScreenCaptureStop()
{
  job.Stop();
}
```

The recorded (joint positions) and calculated (hip angle, knee angle, elbow angle, joint orientation angles) values are saved as a CSV (Comma-separated values) file on the hard disk, whereby each parameter is written in its own file (five files in total). When generating the files in the output folder, a header is inserted. Every time a body frame is captured, the corresponding data are extracted, brought into a readable format and stored in the CSV file. Thus, the files are filled line by line. In all cases the saving process is exactly the same, only the header and the arrangement of the data changes. Additionally, the files for the hip, the knee and the elbow angles consist of a Timestamp with the form "HH:MM:SS.MS". Screenshots of the five generated files with their different headers and values, are pictured in Figure 12**Error! Reference source not found.**.

The general saving process is shown in the code:

```
//Method to insert a new line into an ouput file
public static void CSVFileOutput(String output, String filePath)
{

  //List of strings; each element corresponds to a new line
  List<String> newLines = new List<String>();

  //Load the data "output" into the list
  newLines.Add(output);

  //Write the list into the file at path "filePath"
  File.AppendAllLines(filePath, newLines);

}
```

**Elbow Angles File**

| Time | LeftElbow | RightElbow |
|---|---|---|
| 00:00:00.0000014 | 37 | 37 |
| 00:00:00.3369681 | 37 | 40 |

**Hip Angles File**

| Time | LeftHip | RightHip |
|---|---|---|
| 01:48:00.3397125 | 90 | 3 |
| 01:48:00.3702044 | 90 | 3 |

**Joint Orientation Angles File**

| HipLeft | | | HipRight | | |
|---|---|---|---|---|---|
| X | Y | Z | X | Y | Z |
| 25,34129 | 25,76796 | 22,55718 | 17,41406 | 20,51824 | 21,39896 |
| 25,3404 | 25,75331 | 22,54541 | 17,41375 | 20,51729 | 21,39985 |

**Knee Angles File**

| Time | LeftKnee | RightKnee |
|---|---|---|
| 01:48:00.8693758 | 87 | 43 |
| 01:48:00.9075456 | 88 | 43 |

**Joint Positions File**

| HipLeft | | | HipRight | | |
|---|---|---|---|---|---|
| X | Y | Z | X | Y | Z |
| 0,034347 | 0,063961 | 1,982297 | 0,116571 | 0,082644 | 1,987147 |
| 0,034403 | 0,064092 | 1,982014 | 0,116574 | 0,08269 | 1,98719 |

Figure 12: Screenshots of the generated output files

## One Person Tracking

The Microsoft Kinect$^{TM}$ sensor is capable to detect up to six persons at the same time, but usually the intention is to track only one person. This could lead to problems, such as accidental detection of additional persons (e.g. sessions with trainers) and further to a loss of data due to the additional work of the program. Unfortunately, the "Kinect for Windows Developer Toolkit" doesn't provide any functions to change the number of detectable persons. In order to get the data from only one person some work around has to be done.

The bodies detected by the Kinect$^{TM}$ sensor are saved within an array with six elements, that can be obtained from the body frames with the command `frame.BodyFrameSource.BodyCount`. Each of these bodies have its own identification number (ID), which is randomly generated for a tracked body or zero. The ID number of the first tracked person is memorized within the program, allowing only the data of this number to be executed and ignoring the others. However, if the person is outside the view angle of the Kinect™ the memorized ID number is deleted and a new person can be tracked.

```
//Saved body ID
private ulong ID;

//Called methode when the event fires
public void Reader_MultiSourceFrameArrived(object sender, MultiSourceFrameArrivedEventArgs e)
{

  //Get the frames from the event
  //Get body frame
  //Get the tracked bodies from the body frame

  //auxiliary variables
  int cnt = 0;
  ulong currID = 0;

  //Check if only one body is tracked
  foreach (var body in bodies)
  {
    if (body.TrackingId == 0)    cnt++;

    else    currID = body.TrackingId;
  }

  //Save the ID if only one body is tracked
  if (cnt == 5) ID = currID;

  //For each tracked body
  foreach (var body in bodies)
  {

    //Check if the tracked body holds the saved ID
    if (body.TrackingId == ID)
    {
        //Do Something
    }
  }
}
```

# User Interface of the Program

After the specification of the individual parameters and how they are achieved, the user interface for the program should be discussed. This chapter gives an introduction about the user interface and how the user can communicate with it. Furthermore, it will be explained which selections and settings the user can do in order to optimally adapt the program to the required conditions.

Fundamentally it can be said, that the interface consists of a main window for displaying the images & the output, and two sub windows, which are responsible for the settings. The sub windows are accessed via the main window, which will be described in detail later on.

The main window is the heart of the user interface, from where all settings and selections can be made. It is divided into three areas, with different tasks:

1. A display for viewing the color image and the human body model.
2. A property panel for changing the settings and starting the records.
3. An output grid to display the calculated angles (knee, hip, elbow).

Figure 13 shows a screenshot of the areas of the main window (red border).

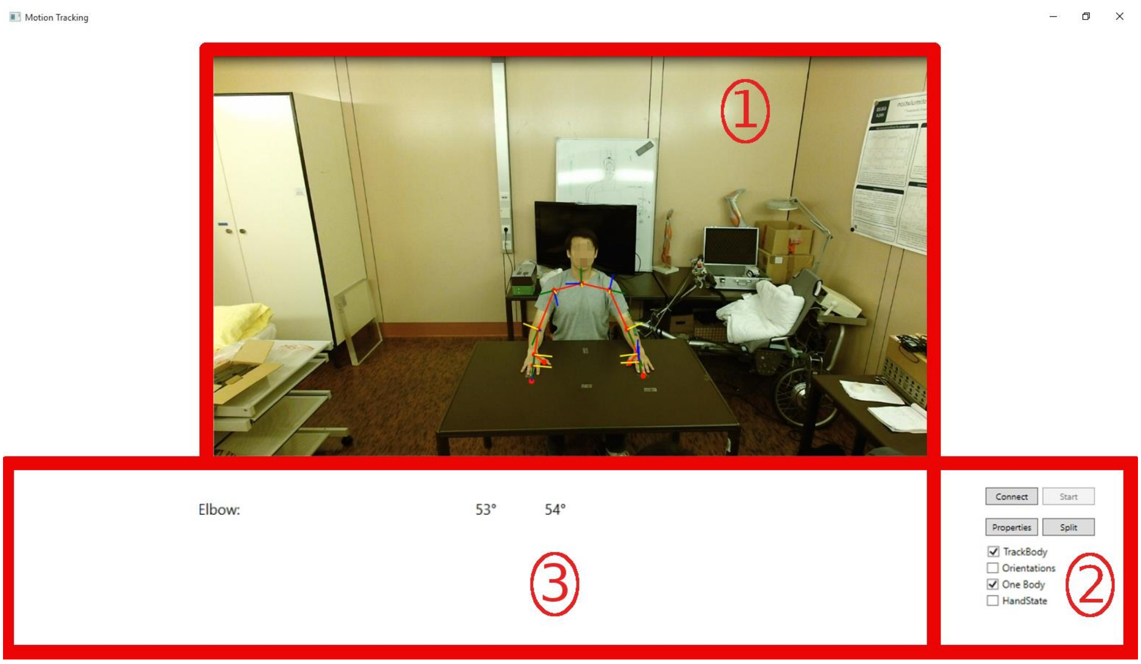



Figure 13: Screenshot of the main window with marked areas

ad 1: The display contains an image which is generated from the color frames provided from the Kinect™ sensor. This image is overlapped with a canvas, on that the human body model, the orientation of the joints and the hand states derived from the body frames are visualized.

ad 2: The property panel consists of check boxes and buttons to change the settings of the display, to choose which parameters are saved onto the hard drive, to create a connection with the Microsoft Kinect™ sensor and to start or end a recording session.

### Connection/Disconnection of the Kinect™ Sensor

The program doesn´t connect to or switch on the camera automatically, it can only be done by clicking the “Connection” button on the panel. This is a safety issue to prevent continuous operation, when it´s not needed (e.g. preparing for a session). Another click on the button disconnects and switches off the camera, which is also automatically done when closing the program. The button gets disabled during a recording session to avoid the appearance of errors.

### Settings for Joints and Angle Calculations

The Microsoft Kinect™ sensor provides the data of 25 joints of the human body (Figure 4). However, often it is desirable to get the data only of the joints of interest (e.g. only legs for gait analysis). This can be done via the “Properties” button, which opens a sub window containing checkboxes for the different joints (Figure 14). They are divided into three columns, where the first one comprises the head to the spine base, the second one the arms and the third one the legs. The checked joints are inserted into an array, that is further used to gain the positions, orientation angles and to draw the corresponding joints onto the canvas as dots. If the array possesses a parent and a child joint, a line is drawn between them representing the bone.

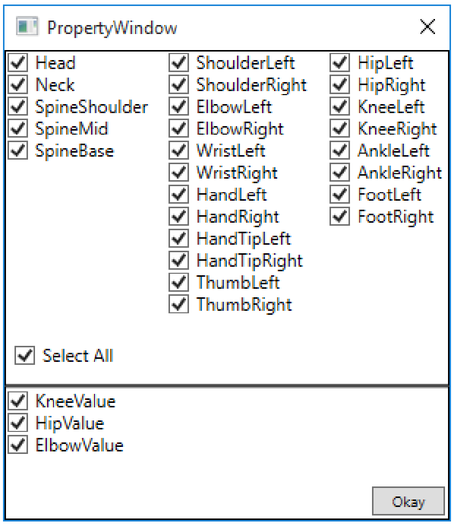



Figure 14: Screenshot of the property window

Additionally, the window consists of two checkboxes for the knee and the hip angle (Figure 14). These two are independent from the settings of the joints, allowing a better control over the single parameters. By checking one of the angles the name is used as an index of a dictionary (“KneeValue”, “HipValue”, “ElbowValue”). During a session the values of the calculated angles are saved in the dictionary according to their respective name, which is further used for saving the data on the hard disk and to display them on the output grid (ad 3). This only works for one person tracking.

The “Properties” button gets disabled during a recording session to prevent the save operation to mix up the joint columns within the file (details on the saving are described in the subchapter “Recording & Calculating the Parameters of Interest”).

### Split/Merge the Image and the Canvas

The image of the color frame is overlapped with the canvas, containing the data from the body frame (Figure 13, area 1). Sometimes the visual view is overloaded with information, especially when a complete human body model is needed. Therefore, the button “Split” was implemented to split up the image and the canvas into two separated parts, by dividing the large area into two smaller areas. One of the areas contains the canvas and the other one the image. Another click on the button reverses the process and merges them back together. This feature is available before and during a recording session.

### Changing the Data drawn on the Canvas

As mentioned before, the job of the canvas is the graphical representation of the joints, the bones, the orientation angles and the state of the hands, that were detected from the Microsoft Kinect™ sensor. With the property panel, the user is able to choose which of these data are displayed on the canvas by using the corresponding checkboxes. Each of these checkboxes works with a Boolean (true/false) allowing or preventing the data to be drawn on the canvas.

The checkboxes are named as follows, first “TrackBody” to activate or deactivate the human body model; second “Orientations” to activate or deactivate the orientation angles and third “HandState” to activate or deactivate the state of the hand. All checkboxes can be set before and during a recording session.

### Enable/Disable One Person Tracking

The program is capable to track only a single person, by ignoring the other ones in the view of the camera. Therefore, people can move freely in front of the Microsoft Kinect™ sensor without influencing the data. The user decides via the checkbox “OneBody” to enable or to disable this function. If unchecked, up to six persons can be tracked, while checking leads to one person tracking. The checkbox can be set before and during a recording session.

### Starting a Recording Session

After the desired settings are made and the Kinect$^{TM}$ sensor is switched on, the recording is initialized with the "Start" button, which opens a sub window (Figure 15) containing three checkboxes and a text field.

If one of the checkboxes is checked, the corresponding data are saved onto the hard disk (video, joint positions, joint orientation angles). Within the text field, the name of the folder where the data are saved has to be entered, otherwise an error message will occur. The "Okay" button starts the recording session.

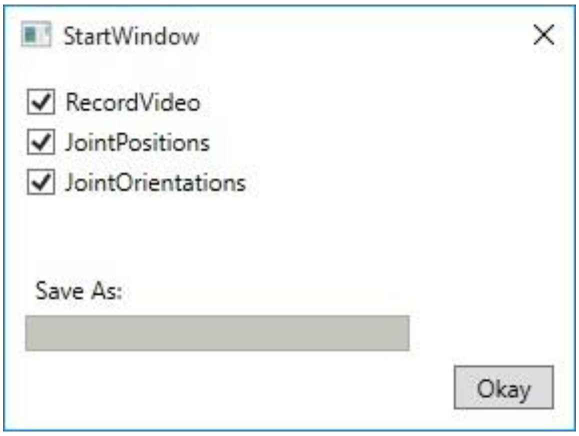



Figure 15: Screenshot of the start window

ad 3: Area 3 consists of an output grid to display the calculated angles in real time. It uses a dictionary, which saves the names of the angles and the corresponding calculated values. Therefore, all entries in the dictionary can be separated into three parts: "name of the joint", "left angle value" and "right angle value".

The grid of the output display is defined as a 4x9 matrix, which allows the displaying of twelve entries ($\frac{4x9}{3}$) in total. Each additional entry is ignored from the grid to avoid "out of bounce" errors. The data are entered in the following format: [name of the joint:] [left angle value °] [right angle value °] and are arranged in a way that is shown in Figure 16.

An example how the data look in the interface can be seen in Figure 13.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 5 | 5 | 5 | 9 | 9 | 9 |
| 2 | 2 | 2 | 6 | 6 | 6 | 10 | 10 | 10 |
| 3 | 3 | 3 | 7 | 7 | 7 | 11 | 11 | 11 |
| 4 | 4 | 4 | 8 | 8 | 8 | 12 | 12 | 12 |

Figure 16: Arrangement of the output grid

# Materials and Methods

In order to prove if the Microsoft Kinect™ sensor provides accurate data for motion tracking and how exact these data are in comparison to conventional methods, data have to be gathered. Therefore, different clinical measurements were performed to collect data, which were subsequently analyzed and evaluated.

The implemented program from the previous chapter was used for the different measurements. A detailed description of the settings for the program and the accomplished measurements are described in their respective subchapters.

## Microsoft Kinect™ Sensor for Gait Analysis

Gait analysis is the study of human walking, it deals with the questions how the human being walks, and what is the proper way to walk (Whittle & Whittle, 2007).

The aim of this measurement was the verification of the quality of the data received from the Microsoft Kinect™ at different walking speed on a treadmill. Furthermore, these data were compared with the data of an electrogoniometer to check their accuracy.

An electrogoniometer is an instrument to measure the angles of a joint. It is attached to the joint in order to track the movement. Thereby, the goniometer produces different voltages depending on its bending, which are recorded and saved by a program.

Two goniometers were placed onto a proband, one on the left hip and the other one on the left knee, to measure the bending of these joints during walking. Parallel to this, the angle of these joints was calculated by the Kinect™.

The task, the proband should accomplish, was walking on a treadmill with different speed. First, a calibration at an angle of 0° and 90° (both hip and knee) was done for the goniometers. Afterwards, the movement was recorded at 1kmph, 2kmph, 3kmph, 5kmph and 8kmph walking speed.

The electrogoniometers, which were used for the measurement were produced from the company BIOMETRICS. They are twin-axis goniometers of the type: **SG150** and are able to measure two rotational axes. One goniometer was placed on the left hip of the proband to get informations about flexion/extension (green connection of the device) and abduction/adduction (grey connection of the device). The other one was positioned on the left knee for flexion/extension (green connection of the device) and varus/valgus (grey connection of the device). An illustration of the placed goniometers is shown in Figure 17.

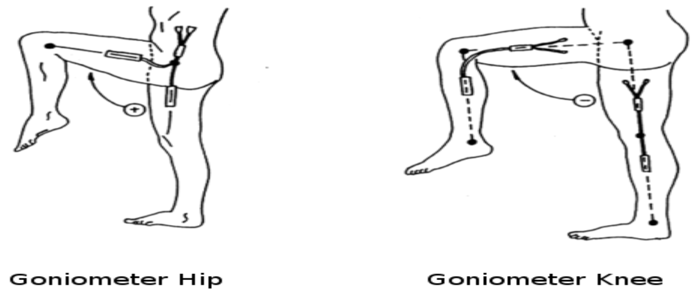



Figure 17: Location of the goniometers (adapted from (Nos, 1998))

The ports of the goniometers were connected with a notebook, which recorded and saved the data via the program DAISYLAB. These data were given as voltage differences over time, which then were converted into degrees [°]. Additional information about the goniometers can be looked up in the data sheet (Nos, 1998).

The Microsoft Kinect™ sensor was positioned laterally to the treadmill, in a way, that the left side of the proband was facing the Kinect™ sensor. This is quite a challenge, because the most accurate recognition is while facing a person and not from the side view (e.g. crossing of the legs while walking). A screenshot from the perspective of the Kinect™ is displayed in Figure 18.

The program had been adjusted in a way, that it only tracks the legs of the proband to save the data, like positions & orientations of the hip, knee, ankle & foot – and the angle of the knees & the hips. Both legs were detected to verify, if they would be effected by the lateral positioning of the Kinect™ sensor.

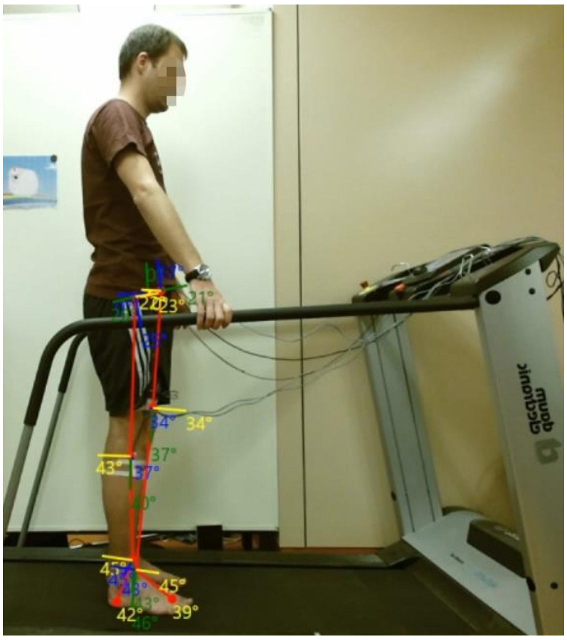


Figure 18: Position of the Kinect™ sensor (mirrored)

### Preparation of the Recorded Data

After the data were recorded, a comparison between the knee and hip angles of the two devices was performed. However, the quality of the data provided from the Microsoft Kinect™ sensor, was investigated beforehand. Thereby it could be seen, that a large amount of values was recognized as undefined, derived from crossing the legs, due to the lateral positioning of the Kinect™ sensor. Because of this crossing it is impossible for the Kinect™ to distinguish between the two legs, leading to undefined values described above. As a consequence, the undefined values got interpolated to allow a comparison with the goniometers. For the interpolation the following rules have been established:

1. If two numbers have the same value, all undefined values between them are filled with the same value.

$$[30\ \ -1\ \ -1\ \ 30]\ \rightarrow [30\ \ 30\ \ 30\ \ 30]$$

2. If undefined values occur between two different numbers, they are filled with ascending or descending values.

$$[30\ \ -1\ \ -1\ \ 90]\ \rightarrow [30\ \ 50\ \ 70\ \ 90]$$

$$[90\ \ -1\ \ -1\ \ 30]\ \rightarrow [90\ \ 70\ \ 50\ \ 30]$$

The unit of the data from the goniometers are given in volt [V], which must first be converted into degree [°]. Therefore, a calibration at 0° and 90° was accomplished for the knee and the hip, in order to achieve the corresponding voltage values. Further, a scaling factor was calculated from these calibration values, allowing to transform all the measured data from the goniometers into degree [°].

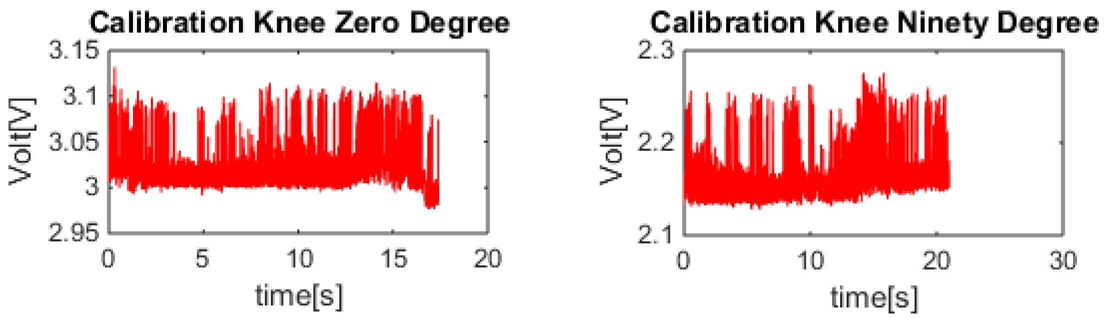



Figure 19: Recorded values for the calibration of the knee

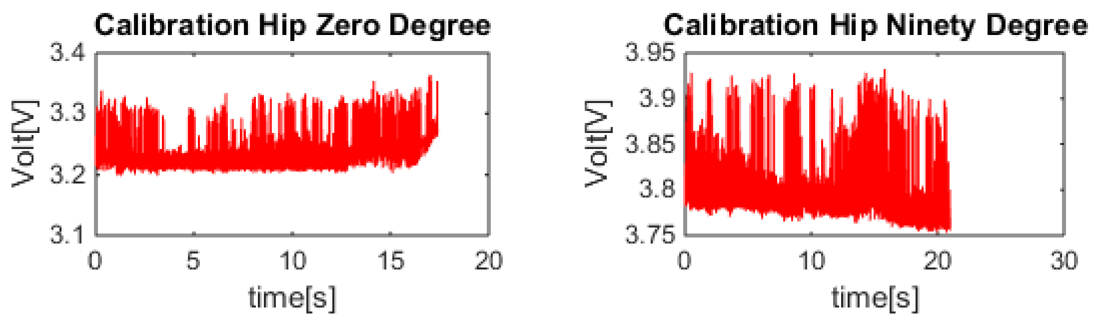



Figure 20: Recorded values for the calibration of the hip

Figure 19 and Figure 20 show the recorded voltages for the knee and the hip at 0° and 90°, which was used to calculate the scaling factors. From these data the arithmetic mean was determined, followed by the linear correlation between 0° and 90° of the according joints:

$$mean\ knee\ 0° = 3.02V \qquad mean\ hip\ 0° = 3.23V$$

$$mean\ knee\ 90° = 2.16V \qquad mean\ hip\ 90° = 3.79V$$

$$k = \frac{3.02V - 2.16V}{0° - 90°} = -9.6mV/° \qquad k = \frac{3.23V - 3.79V}{0° - 90°} = 6.2mV/°$$

$$\mathrm{y} = \frac{3.02\mathrm{V} - \mathrm{x}}{-9.6\mathrm{mV}/°} \qquad \mathrm{y} = \frac{3.23\mathrm{V} - \mathrm{x}}{6.2\mathrm{mV}/°}$$

where x is the recorded voltage [V] and y its corresponding value in degree [°].

# Microsoft Kinect™ Sensor for Hand Motion & Grasping

Starting from an initial position the proband had to fulfill different hand motion and grasping tasks, which were divided into three categories: “full range of motions”, “arm crossing” and “grasping an object”. The proband was sitting in a chair and not allowed to move the upper body except the arms. Each task was started and stopped with the same initial hand position, in which both hands were laid on a predefined location, so that the elbows were angled 50°-60°.

The aim of each category was to examine if the Microsoft Kinect™ sensor is able to track the performed motion or gesture. Further the accuracy of the obtained data was investigated by comparing them with the data from an electrogoniometer.

An electrogoniometer is an instrument to measure the angles of a joint. It gets attached to the joint in order to track the movement. Thereby, the goniometer produces different voltages depending on its bending, which are recorded and saved by a program.

Two goniometers were placed onto the proband, one on the right shoulder and the other one on the right elbow, to measure the bending of these joints during the tasks. Parallel to this, the angle and the positions of these joints got calculated by the Kinect™ sensor.

In the categories “full range of motions” and “arm crossing” the tasks of the proband were to posture in different prescribed ways. Starting from an initial hand position the proband strikes the pose for five seconds and returns back into the initial position. The poses for “full range of motions” were chosen to prefer one axes of the three dimensional space (Figure 21, left), while for “arm crossing” the arms are crossed in front of the body and above the head (Figure 21, right).

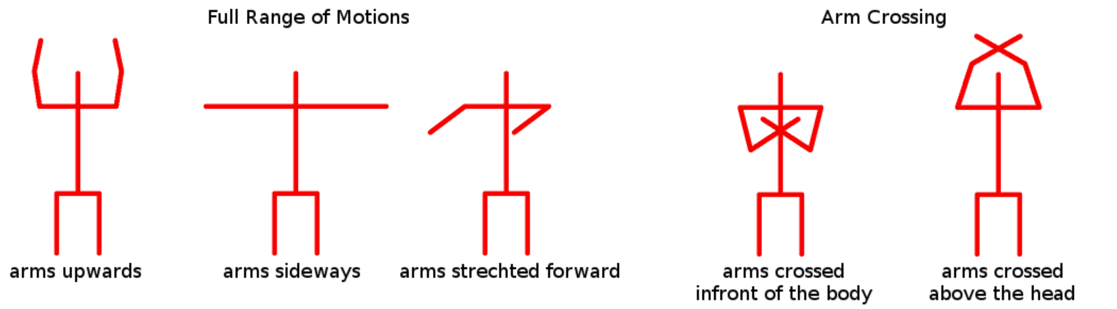



Figure 21: Hand motion tasks "full range of motions" & "arm crossing"

In comparison to that, the third category “grasping an object” was about movement detection. The proband had to move a particular object (e.g. bottle, cup, …) from position A to position B and back again, using the right arm (Figure 22).

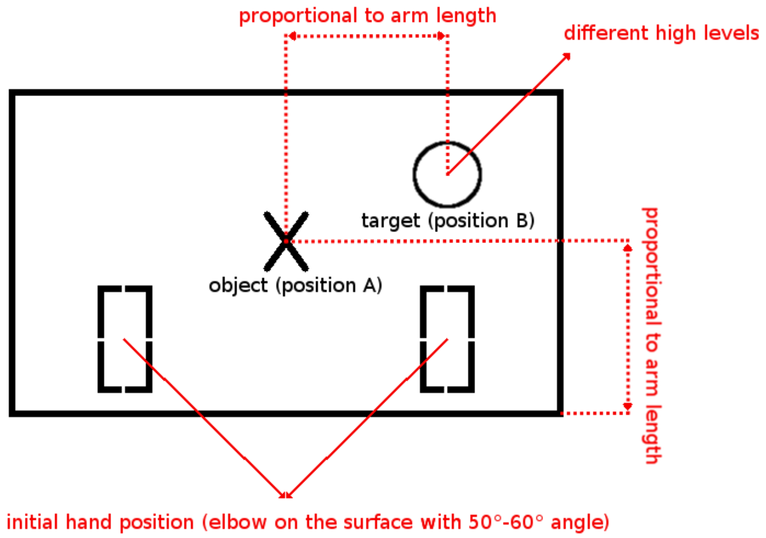



Figure 22: Top view of the desk for the task “grasping an object”

A complete cycle of a grasping task was defined as follows: Proceeding from the initial hand position the proband moves an object with the right arm from position A to position B and then returns the arm back to the starting position. After a short period, the proband moves the object back to position A. This was repeated five times. Differences in the tasks were depending on the used object (bottle, cup) and the hight level of the target (position B, at 0 cm or 15 cm), leading to four varied tasks in total. The distance between the positions, as well as the distance between the proband and the positions was chosen corresponding to the length of the arm (Figure 22).

### Experimental Setup

The proband, who performed the tasks was 176 centimeter (cm) tall. From that the location for the initial hand position, the position of the object (position A), the position of the target (position B) and the different hight levels for position B were calculated (Figure 23). The objects used for the "grasping an object" tasks were a bottle and a cup.

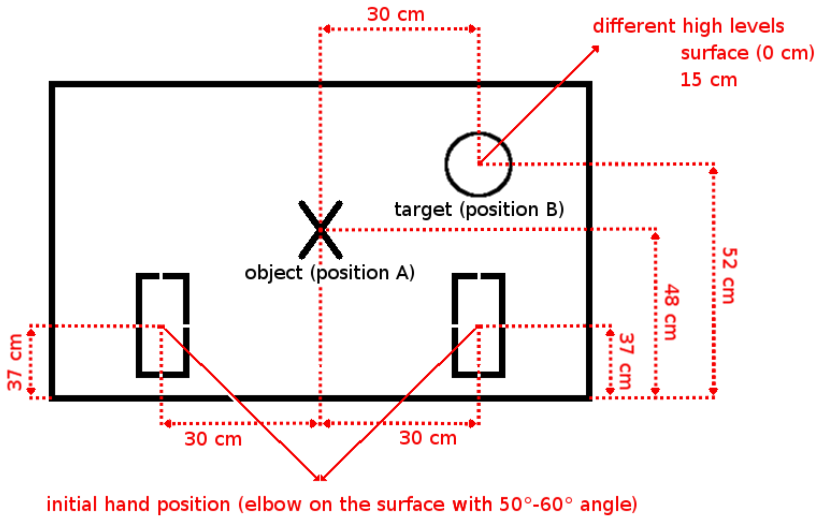



Figure 23: Top view of the desk with calculated values

Further, this measurement was repeated three times under different conditions: First, the proband was sitting behind a desk and the Microsoft Kinect™ sensor was positioned in front of the proband and at a higher level (Figure 24, left). Second, the desk was turned 30°- 45°, so that the right side of the proband was closer to the Kinect™ sensor compared to the rest of the body (Figure 24, mid). Third, the proband was facing the Kinect™ sensor similar to the first condition, but without a desk (Figure 24, right).

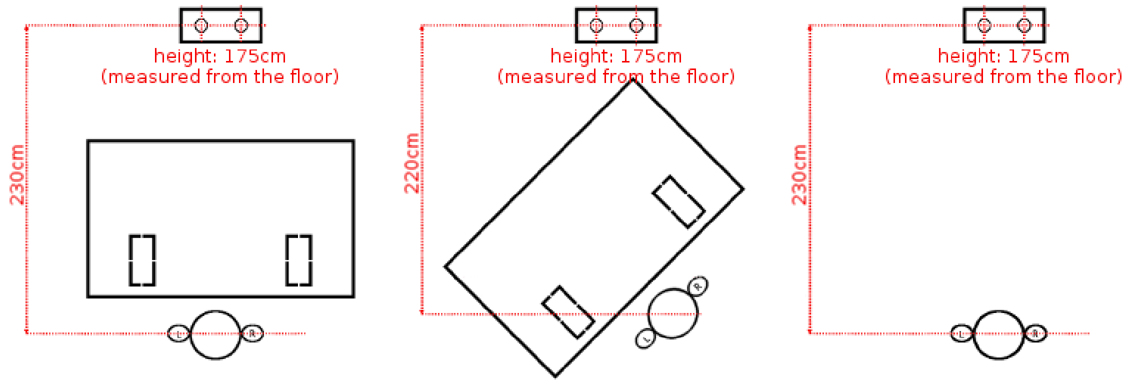



Figure 24: Conceptual design of the three measurements

The program for the Kinect™ had been adjusted in a way, that it only tracks the arms of the proband to save the data, such as positions & orientations of the spine shoulder, shoulder, elbow, wrist, hand, hand tip & thumb – and the angle of the elbows. A screenshot from the perspective of the Kinect™ sensor during the initial hand position for the three different conditions is displayed in (Figure 25).

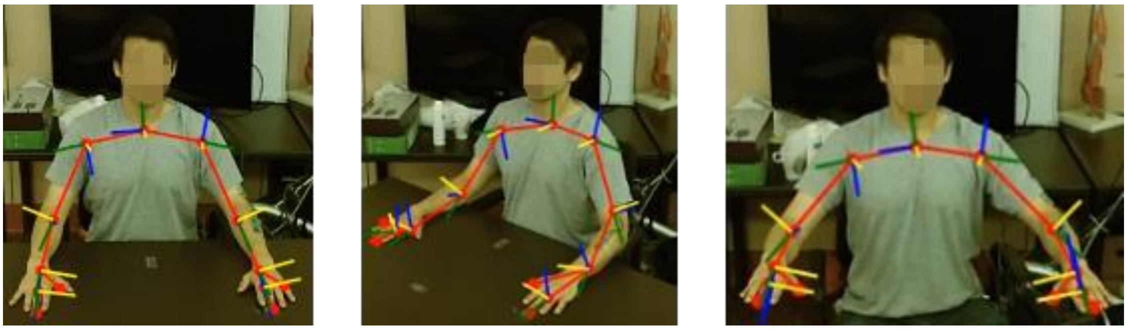


Figure 25: Initial hand position during different conditions

The electrogoniometers which were used for the measurement were produced from the company BIOMETRICS. They are twin-axis goniometers of the type: **SG150** and are able to measure two rotational axes. One goniometer was placed on the right elbow of the proband to get informations about flexion/extension (green connection of the device). The other one was positioned on the right shoulder for flexion/extension (green connection of the device) and abduction/adduction (grey connection of the device). An illustration of the placed goniometers is shown in Figure 26.

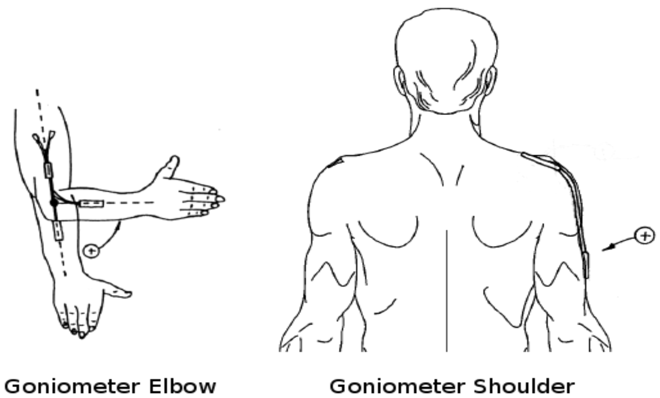


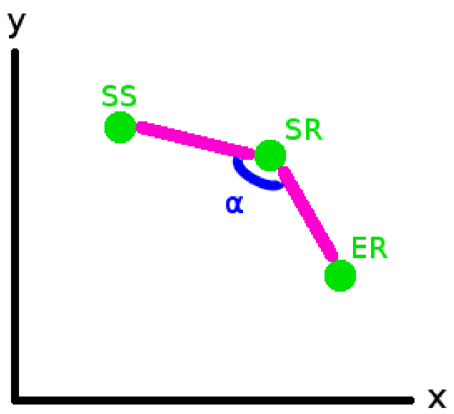

Figure 26: Location of the goniometers (adapted from (Nos, 1998))

The ports of the goniometers were connected with a notebook, which recorded and saved the date via the program DAISYLAB. These data were given as voltage differences over time, which got further be converted into degrees [°]. Additional information about the goniometers can be looked up in the data sheet (Nos, 1998).

### Preparation of the recorded data

In advance of the comparison between the data of the goniometer and the Microsoft Kinect™ sensor some preparations were done. First, the angles of the right shoulder were calculated from the three dimensional positions provided from the Kinect™ sensor. Second, the data from the electrogoniometer were converted from voltage into degree. Third, for both devices the initial hand position was chosen as starting position, at which all angles (elbow, shoulder) were set relatively to 0°.

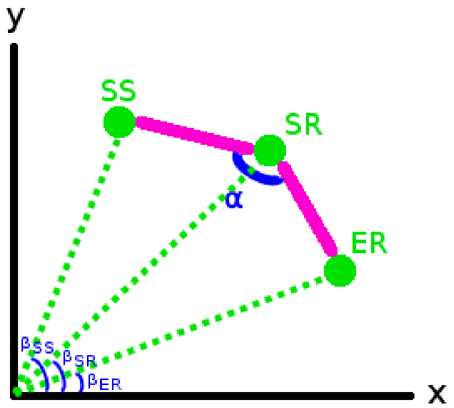
To achieve the angles of the shoulder, the three dimensional coordinates of the joints Spine Shoulder (SS), Shoulder Right (SR) and Elbow Right (ER) are required. From that, the vectors for the upper arm (UA = ER - SR) and the shoulder (S = SR - SS) can be calculated. The angle (α) between these two vectors should be determined for each plane (xy, xz, yz). The procedure is explained with reference to the xy plane and is analogous for the other two.




In the first step, a translation is done in order to move the points of the joints into the first quadrant of the coordinate system:

$$\overrightarrow{SS} = \overrightarrow{SS} + t, \qquad \overrightarrow{SR} = \overrightarrow{SR} + t, \qquad \overrightarrow{ER} = \overrightarrow{ER} + t$$

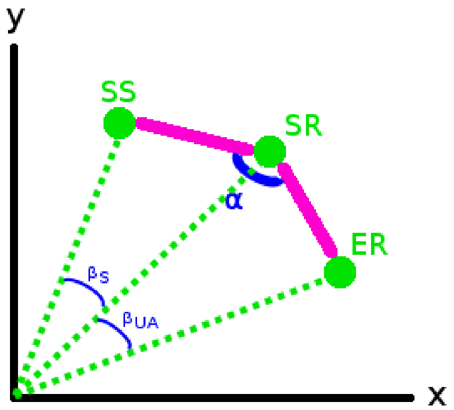
where t represents the translation value.




Next, the angle between the three points and the abscissa is calculated using the formula:

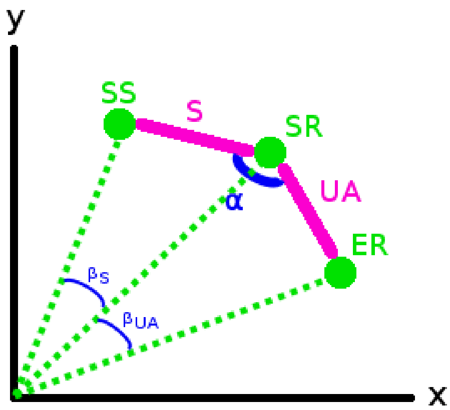
$$\beta = \mathrm{acos}(\frac{v_x}{\bar{v}}),$$

where β is $\beta_{SS}$, $\beta_{SR}$ & $\beta_{ER}$; $v_x$ is the x component and $\bar{v}$ the length of the xy – projection of the points SS, SR & ER.




With the obtained angles ($\beta_{SS}$, $\beta_{SR}$, $\beta_{ER}$) the enclosed angles ($\beta_S$, $\beta_{UA}$) between the points SS & SR and SR & ER, are given as:

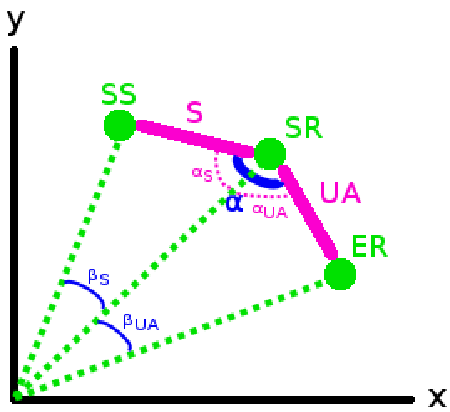
$$\beta_S = \beta_{SS} - \beta_{SR}, \qquad \beta_{UA} = \beta_{SR} - \beta_{ER}$$




Using the law of cosine, the length of the shoulder (S) and the length of the upper arm (UA) can be calculated:

$$\bar{S} = \overline{SS}^2 * \overline{SR}^2 - 2 * \overline{SS} * \overline{SR} * \cos(\beta_S),$$

$$\overline{UA} = \overline{SR}^2 * \overline{ER}^2 - 2 * \overline{SR} * \overline{ER} * \cos(\beta_{UA})$$




The desired angle α consists of two pitch angles ($\alpha_S$, $\alpha_{UA}$), which can be described with the law of sine:

$$\alpha_S = \operatorname{asin}\left(\frac{\sin(\beta_S)}{\bar{S}} * \overline{SS}\right),$$

$$\alpha_{UA} = \operatorname{asin}\left(\frac{\sin(\beta_{UA})}{\overline{UA}} * \overline{SS}\right)$$

The sum of these two vectors is α:

$$\alpha = \alpha_S + \alpha_{UA}$$

In comparison to that, the right elbow was recorded throughout the measurement with the program, so that it values only needed to be altered in order to set the initial hand position as starting position. Therefore, the angle at the initial hand position was subtracted from all the measured data points. Additionally, the angles of the projections of the right elbow for the xy-, xz-, yz- plane was calculated with the same approach as it was done for the shoulder. The used three dimensional coordinates derived from the points: Shoulder Right, Elbow Right & Wrist Right.

Finally, the voltage data from the goniometers had to be transformed into degree [°] and adapted to the initial hand position. A scaling factor was calculated using the voltage values at the initial hand position and the voltage values from the second task of the category "full range of motions", leading to the equations:

$$mean\ shoulder\ flex\ 25° = 2.83V$$

$$mean\ shoulder\ flex\ 0° = 3.08V$$

$$k = \frac{2.83V - 3.08V}{25° - 0°} = -10mV/°$$

$$\mathrm{y} = \frac{2.83\mathrm{V} - \mathrm{x}}{-10\mathrm{mV}/°}$$

$$mean\ shoulder\ abd\ 5° = 3.05V$$

$$mean\ shoulder\ abd\ 90° = 2.72V$$

$$k = \frac{3.05V - 2.72V}{5° - 90°} = -3.9mV/°$$

$$\mathrm{y} = \frac{3.05\mathrm{V} - \mathrm{x}}{-3.9\mathrm{mV}/°}$$

$$mean\ elbow\ flex\ 55° = 2.71V$$

$$mean\ elbow\ flex\ 0° = 3.23V$$

$$k = \frac{2.71V - 3.23V}{55° - 0°} = -9.5mV/°$$

$$\mathrm{y} = \frac{2.71\mathrm{V} - \mathrm{x}}{-9.5\mathrm{mV}/°}$$

# Results

The results obtained from the performed measurements (gait analysis, hand motion & grasping), which were described in the previous chapter, will be illustrated and explained below.

## Microsoft Kinect™ Sensor for Gait Analysis

The aim of this measurement was the verification of the quality of the data received from the Microsoft Kinect™ and to compare them with the data of a goniometer. Therefore, data were collected from the left knee angle and the left hip angle at different walking speed (1kmph, 2kmph, 3kmph, 5kmph, 8kmph) using both devices. Figure 27 shows the data of the knee and the hip at a speed of 3kmph, where the blue graph represents the data from the Kinect™ and the red graph represents the data from the goniometer.

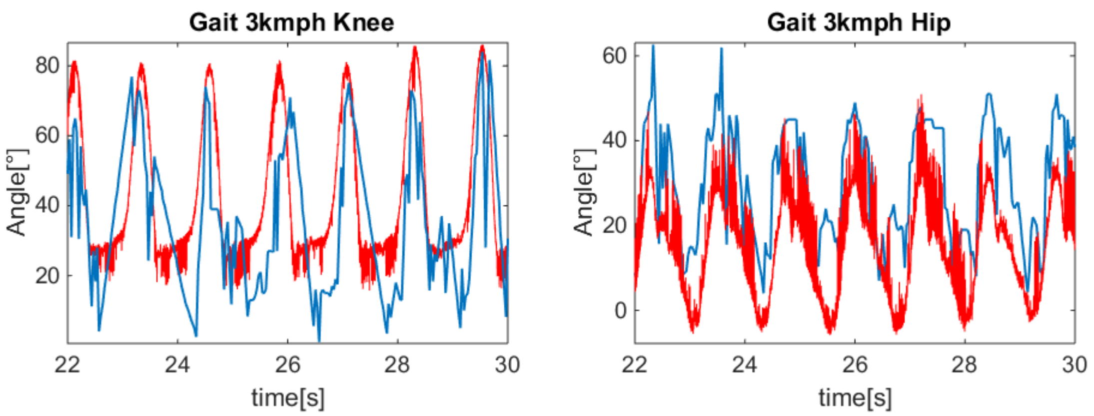
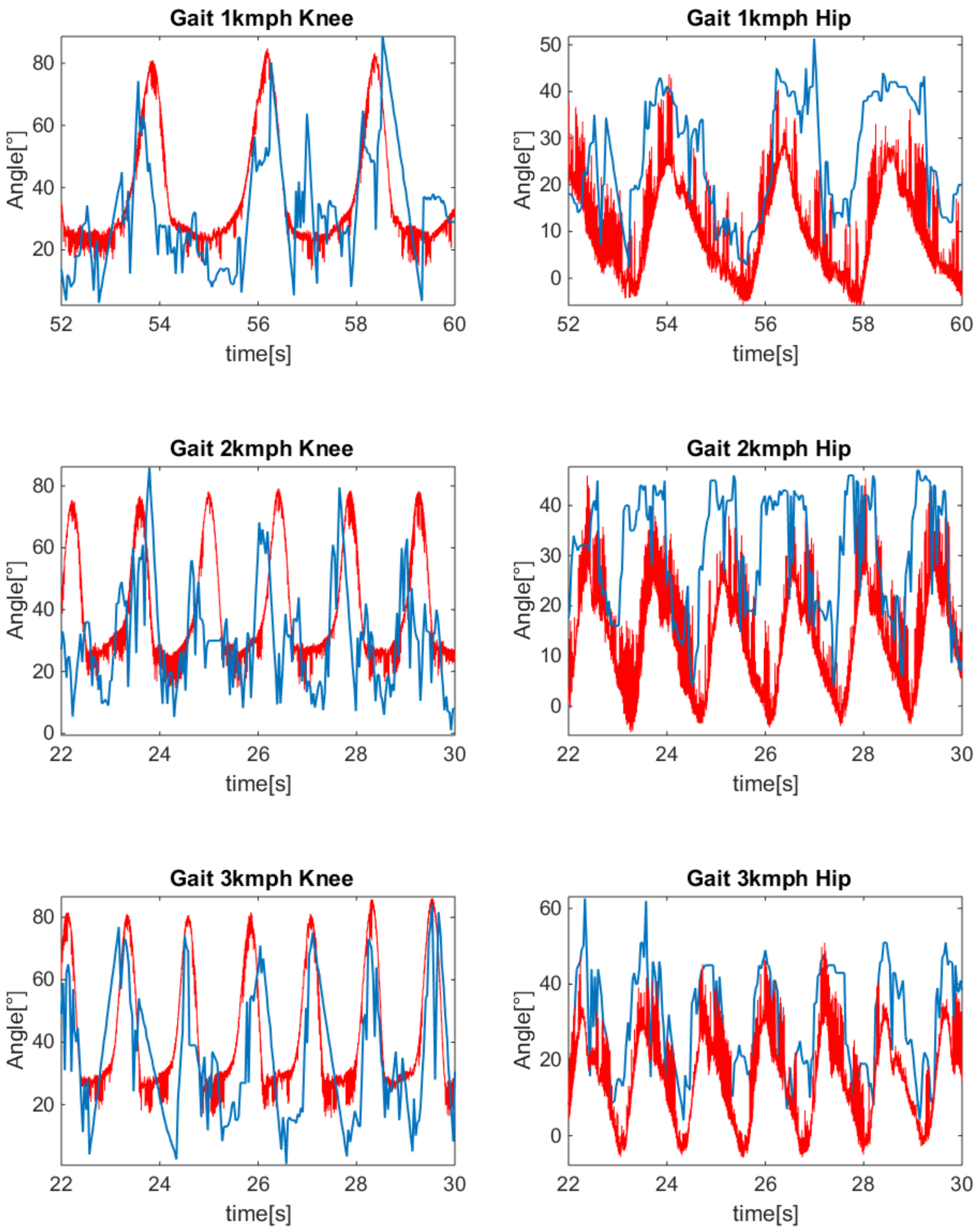



Figure 27: Gait analysis, plotted knee and hip at 3kmph (zoom)

The recorded graphs look similar for both the knee and the hip, but taking a closer look reveals, that the blue graph is missing important values, leading to gaps and incorrect recognition of the peaks (Figure 27).

Another problem is the shape of the peaks (Figure 27). While the shape of the peaks from the goniometer looks relatively similar to each other, it differs within the data of the Kinect™.

The problems explained above occur in all of the recorded data, regardless of the walking speed and joint. That's why all the remaining data will be displayed for completeness, but not be described any further (Figure 28).



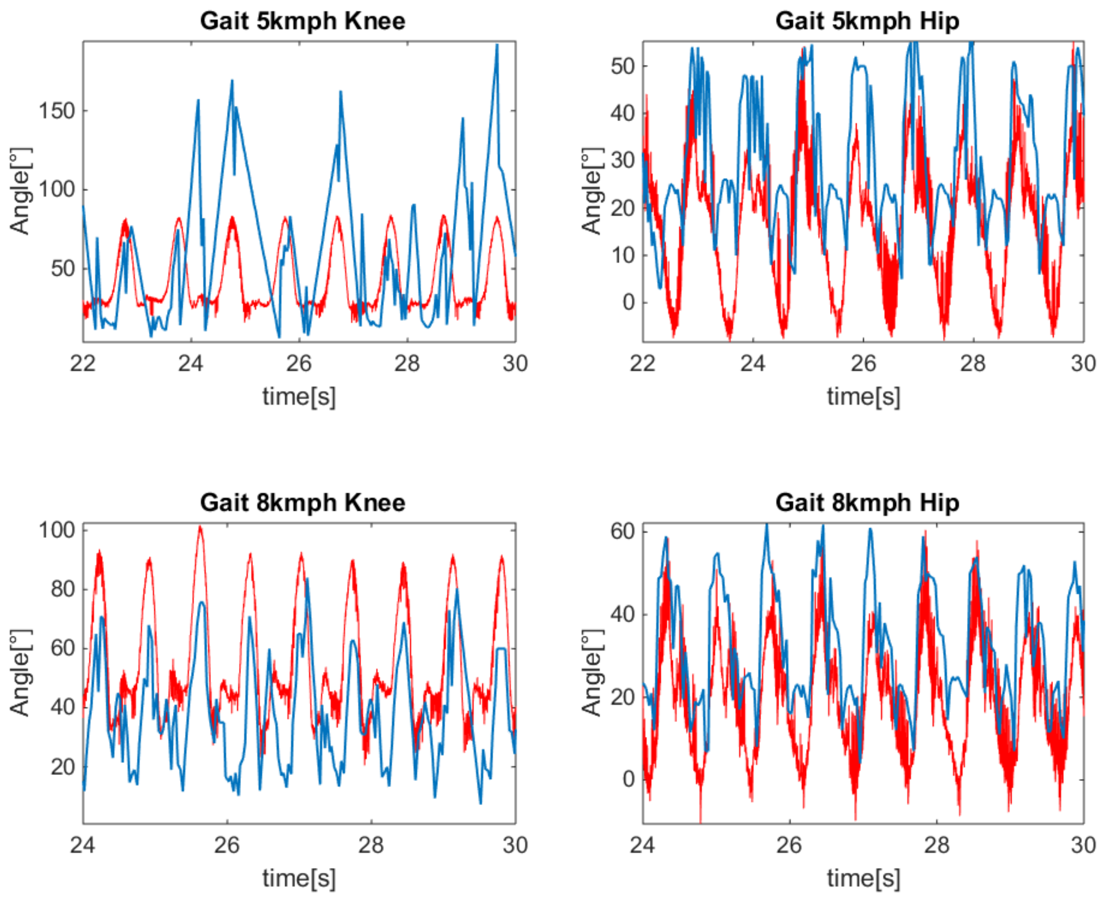



Figure 28: Gait analysis, knee & hip angle at different walking speed

## Microsoft Kinect™ Sensor for Hand Motion & Grasping

Different tasks were performed, divided into three categories: "full range of motions", "arm crossing" and "grasping an object". In each of these categories the right elbow and the right shoulder were recorded. The measurement was repeated three times under different conditions (proband in front of the Kinect™, proband 30°-45° turned to the Kinect™ & proband in front of the Kinect™ without any objects).

Recording the data of the Kinect™ sensor was accomplished with a resolution of 30 frames per second, while the resolution of the goniometers were 1000 data points per second.

The red graph represents the data from the goniometers and the blue graph the angles calculated from the three dimensional coordinates of the Microsoft Kinect™ sensor.

## RESULTS OBTAINED FROM THE SHOULDER

“Full range of motions” includes three tasks in which the arms of the proband were stretched upward, lateral and forward.

### Arms stretched upward

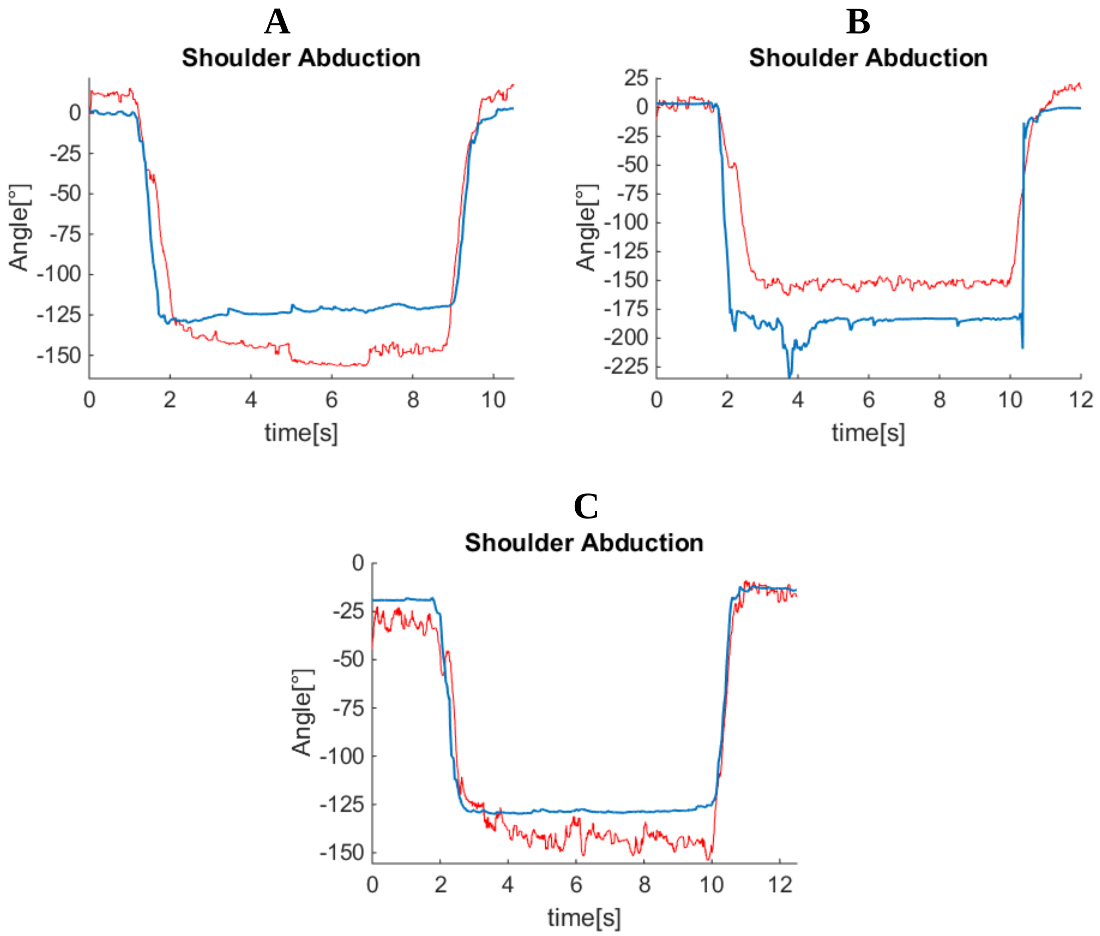



Figure 29: Shoulder abduction from “full range of motions”, arms upward, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

The abduction of the second measurement (Figure 29, B) contains higher negative values and a negative peak compared to the others. Within the first measurement (Figure 29, A) a slight increase of the angle, during the stretching of the arms can be noticed. Also, the starting point of the third measurement (Figure 29, C) is not around 0°, like the others, but lower (~ -25°).

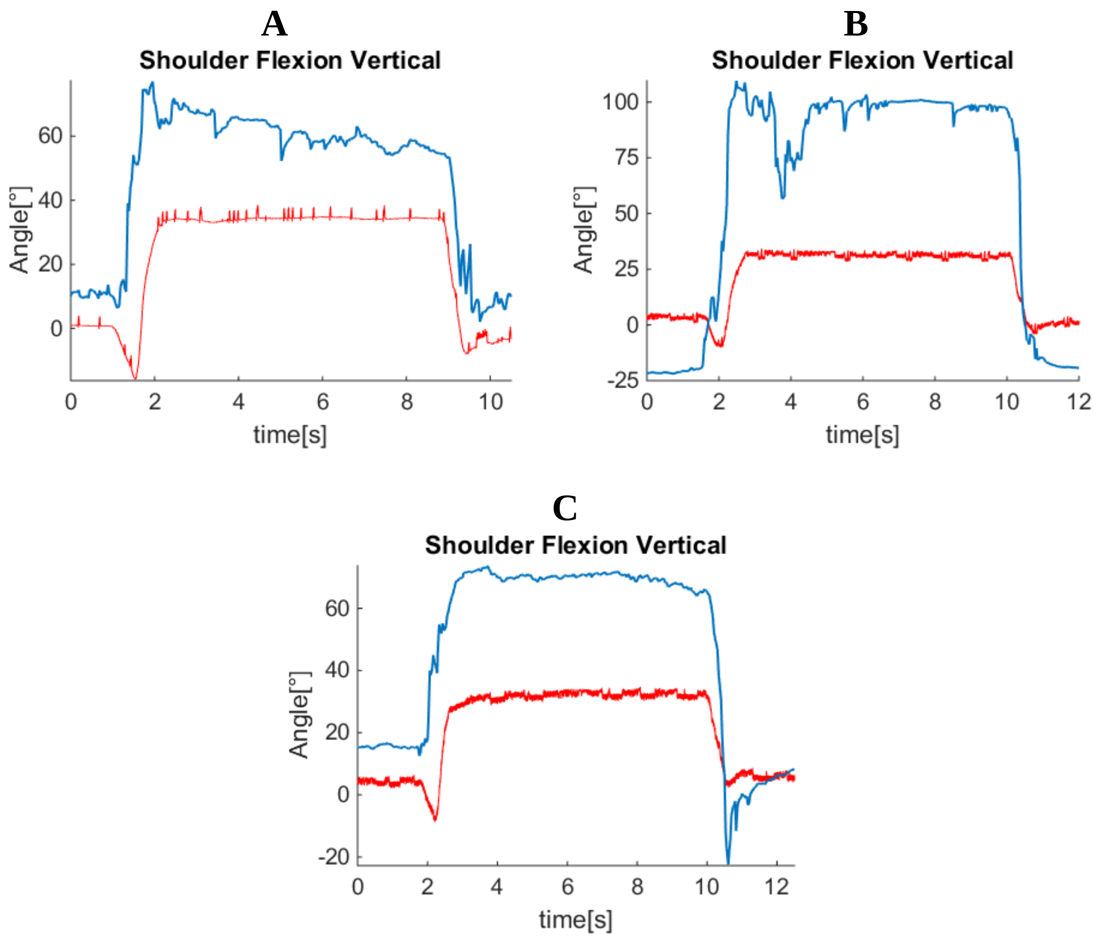



Figure 30: Shoulder vertical flexion from "full range of motions", arms upward, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

The angle of the second measurement (Figure 30, B), as it could already be seen at the abduction, consists of a substantially higher angle than the other two measurements (Figure 30, A, C), and a negative peak. Further, irregularities and dropping values appear in the graph of the first measurement (Figure 30, A) compared to the third measurement (Figure 30, C). Also the starting points of all three measurement differ from 0°. The first & third measurement (Figure 30, A, C) have a similar starting point (~15°), while the starting point of the second one differs (~ -20). During the end of the third measurement (Figure 30, C) a negative peak is noticeable, which doesn't appear in the other measurements (Figure 30, A, B).

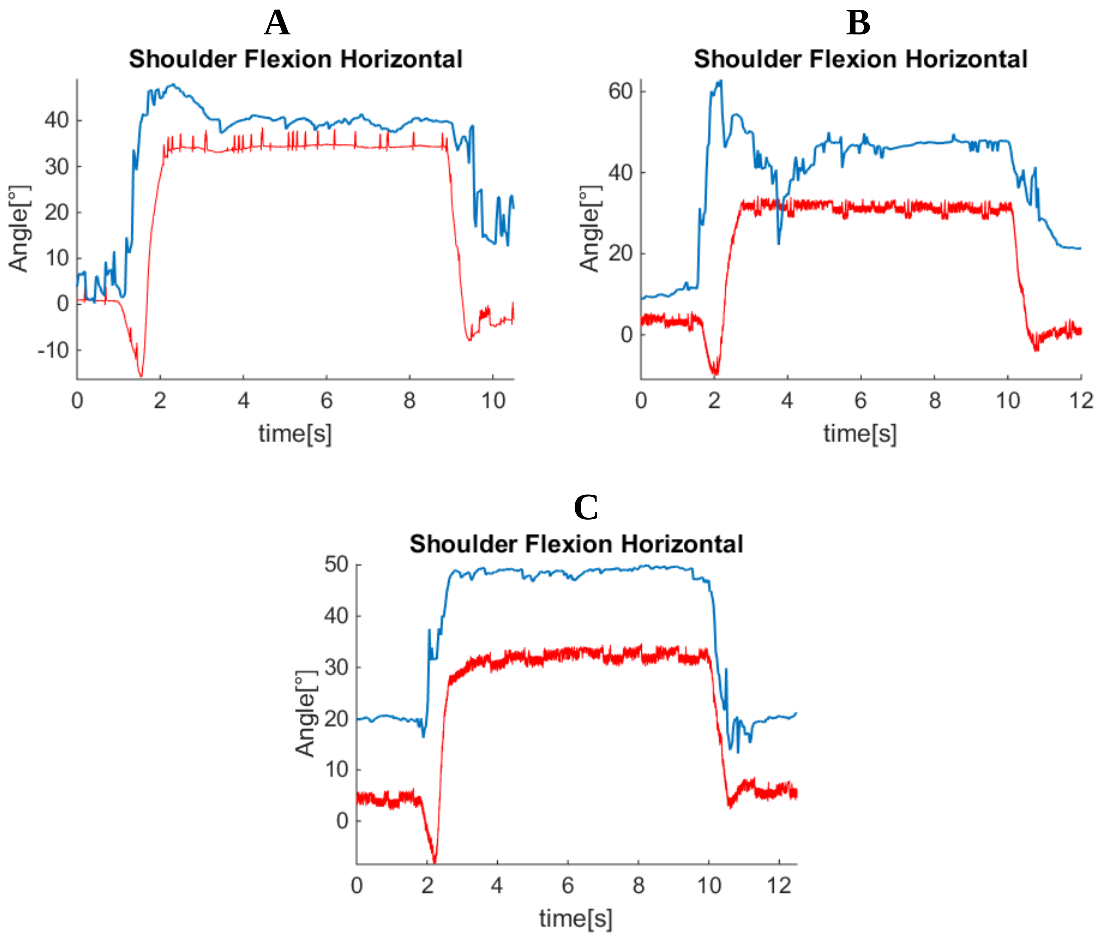



Figure 31: Shoulder horizontal flexion from "full range of motions", arms upward, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

As it was already seen in the abduction & vertical flexion, a negative peak occurs within the second measurement (Figure 31, B), it also has the largest maximum angle of all three measurements. Closer examination reveals, that the angle in the first & second measurement (Figure 31, A, B) is dropping and jumping while stretching the arms, in comparison to the third measurement (Figure 31, C), which remains fairly constant. In addition, the angle at the end of the first & second measurement (Figure 31, A, B) is not equal to the starting angle, as it is for the third measurement (Figure 31, C). Also, the starting angle of the first & second measurement (Figure 31, A, B) is approximately 0°, while it is around 20° for the third one (Figure 31, C). At the beginning and at the end of the first measurement (Figure 31, A) the angle is very jumpy, which also occur at the end of the third measurement (Figure 31, C).

Arms stretched lateral

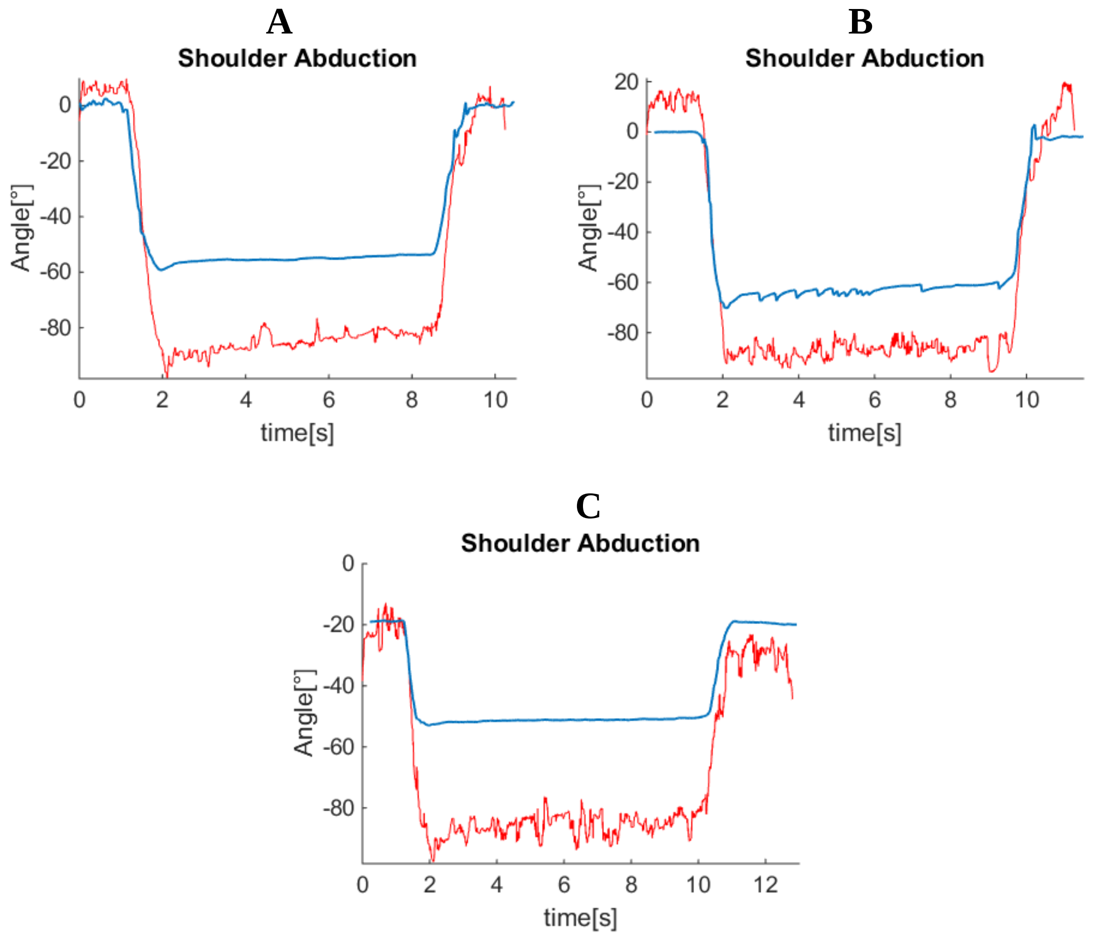



Figure 32: Shoulder abduction from "full range of motions", arms lateral, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

When the arms are stretched laterally, the second measurement (Figure 32, B) shows the largest maximum angle, however, the maximum angle in the first and third measurement (Figure 32, A, C) is similar, but the starting value is different. The initial value of the first & second measurement (Figure 32, A, B) is around 0°, while at the third one (Figure 32,

C) it is around -20°. Also, small peaks occur during the stretching of the arms within the second measurement (Figure 32, B), but not in the first & third one (Figure 32, A, C).

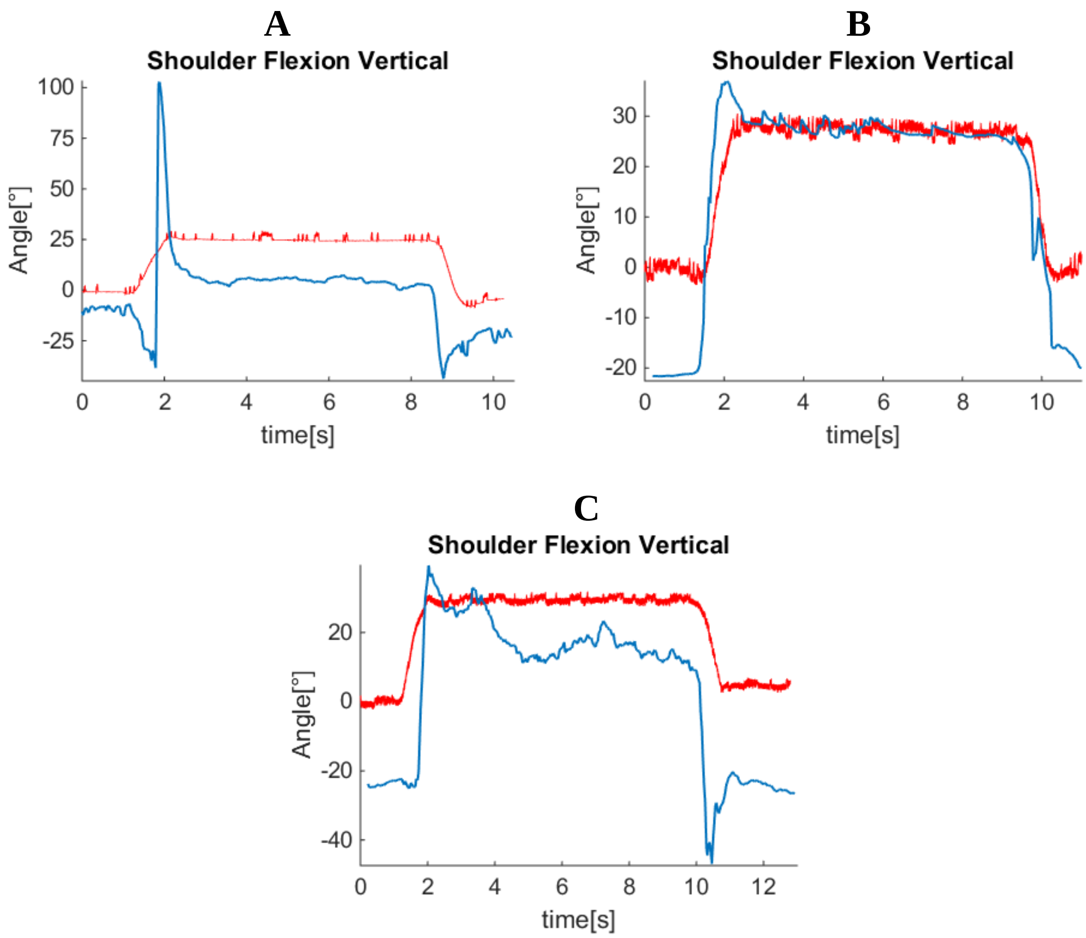



Figure 33: Shoulder vertical flexion from "full range of motions", arms lateral, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

Within the first measurement (Figure 33, A) a large positive peak is noticeable, which also occurs in the second & third measurement (Figure 33, B, C), but much smaller. Additionally, the first measurement (Figure 33, A) contains two negative peaks at the transition from the initial hand position into the stretching and vice versa, furthermore, one negative peak occurs at the end of the third measurement (Figure 33, C), but no peaks arise in the second one (Figure 33, B). The measured angle from the first measurement (Figure 33, A) hardly changes from 0°, compared to the second & third measurement (Figure 33, B, C). Within the third measurement (Figure 33, C) irregularities and dropping values appear during the stretching, compared to the first & second measurement (Figure

33, A, B), where the graph is relatively constant. The initial angle from the second & third measurement (Figure 33, B, C) is approximately -20°, while the initial value for the first measurement (Figure 33, A) is around -10°.

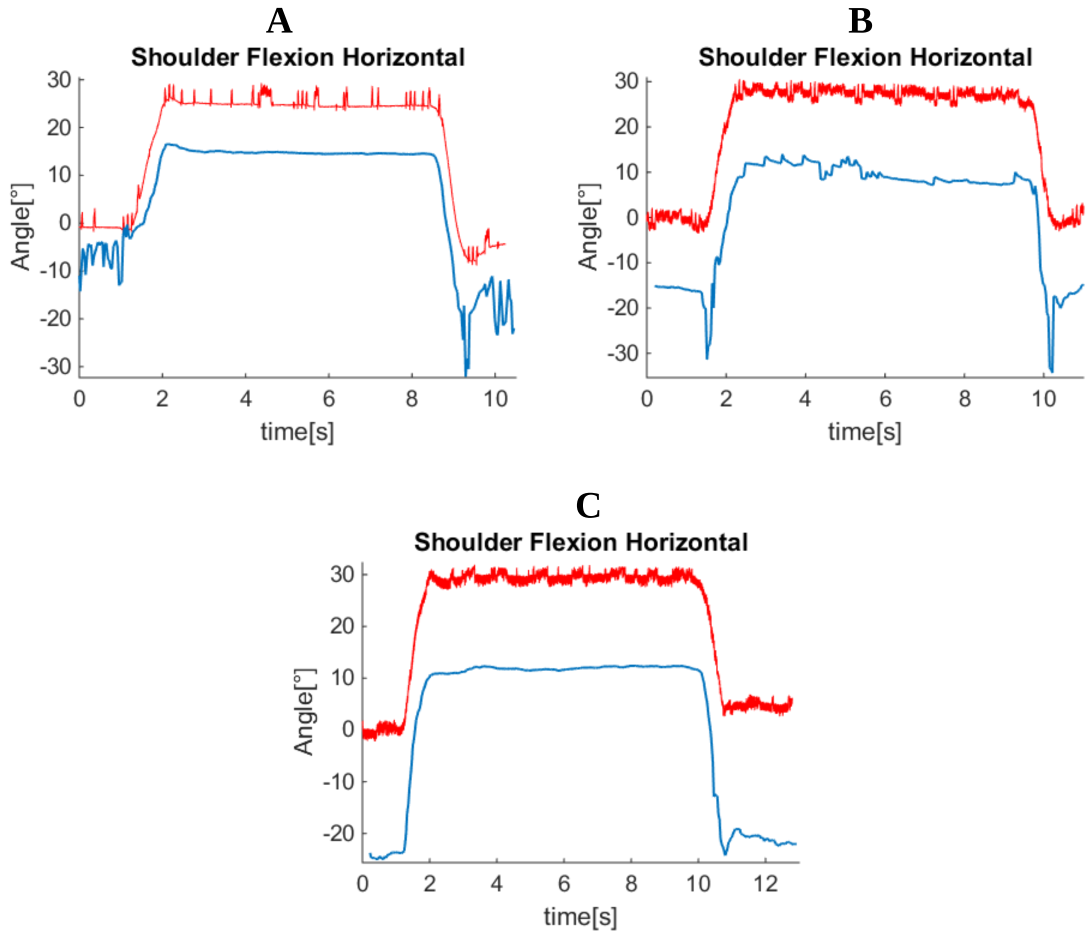



Figure 34: Shoulder horizontal flexion from "full range of motions", arms lateral, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

At the initial hand positions jumpy values can be seen in the first measurement (Figure 34, A), while negative peaks arise at these points in the second measurement (Figure 34, B), compared to the third one (Figure 34, C) where only a small negative peak occurs. In addition, the angle in the second measurement (Figure 34, B) is jumpy during the stretching process, which is not the case in the first & third measurement (Figure 34, A, C), where the angle remains constant. For all three measurements (Figure 34) the starting value differs from each other, first measurement (Figure 34, A) has ~-10°, second one (Figure 34, B) has ~-15° and the third one (Figure 34, C) has ~-25°.

Arms stretched forward

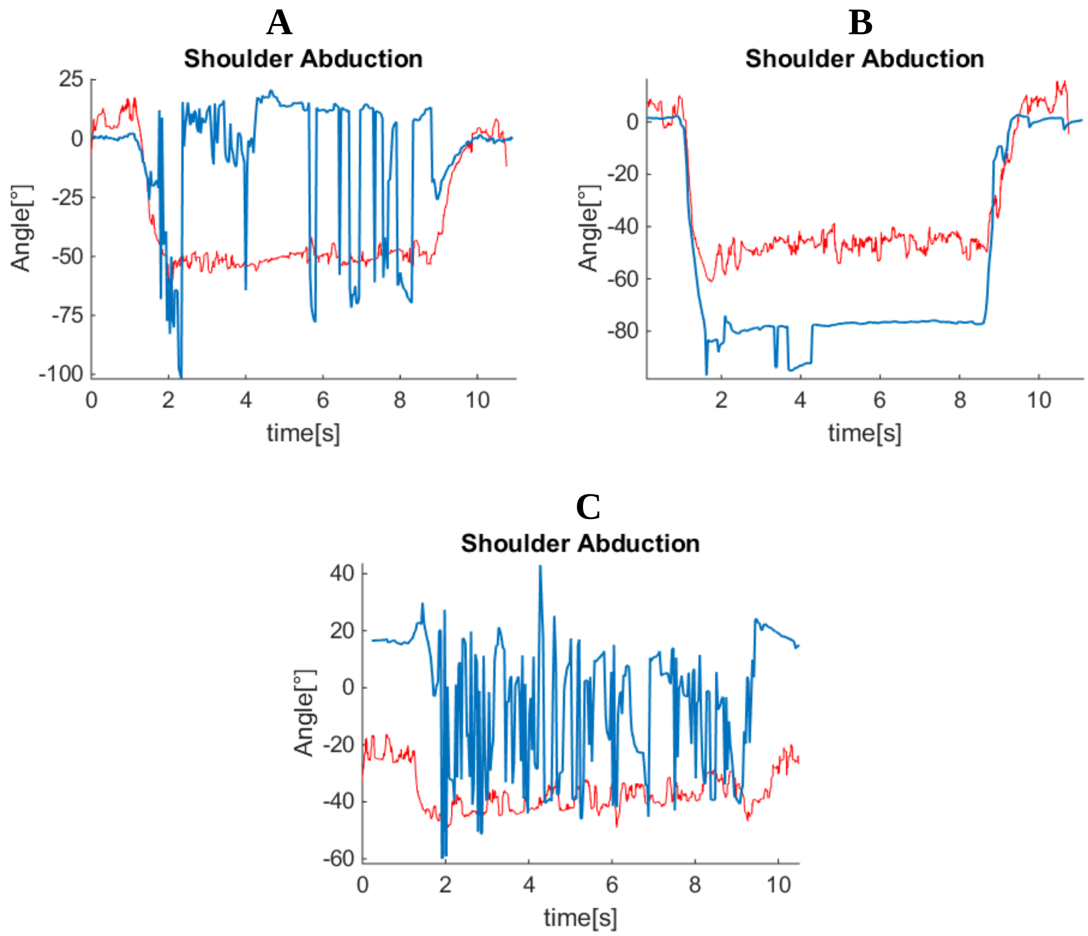



Figure 35: Shoulder abduction from "full range of motions", arms forward, Kinect$^{TM}$ in front with desk (A), desk turned 30°-45° (B), Kinect$^{TM}$ in front no desk (C)

The data from the first & third measurement (Figure 35, A, C) are very jumpy and irregular, although it should be said, that the results from the first measurement (Figure 35, A) are not as pronounced as the data from the third one (Figure 35, C). Quite different are the results of the second measurement (Figure 35, B), these are relatively constant and contain only three negative peaks. Additionally, the third measurement (Figure 35, C) has a starting value around 20°, while the starting point for the first & second measurement (Figure 35, A, B) is nearly 0°.

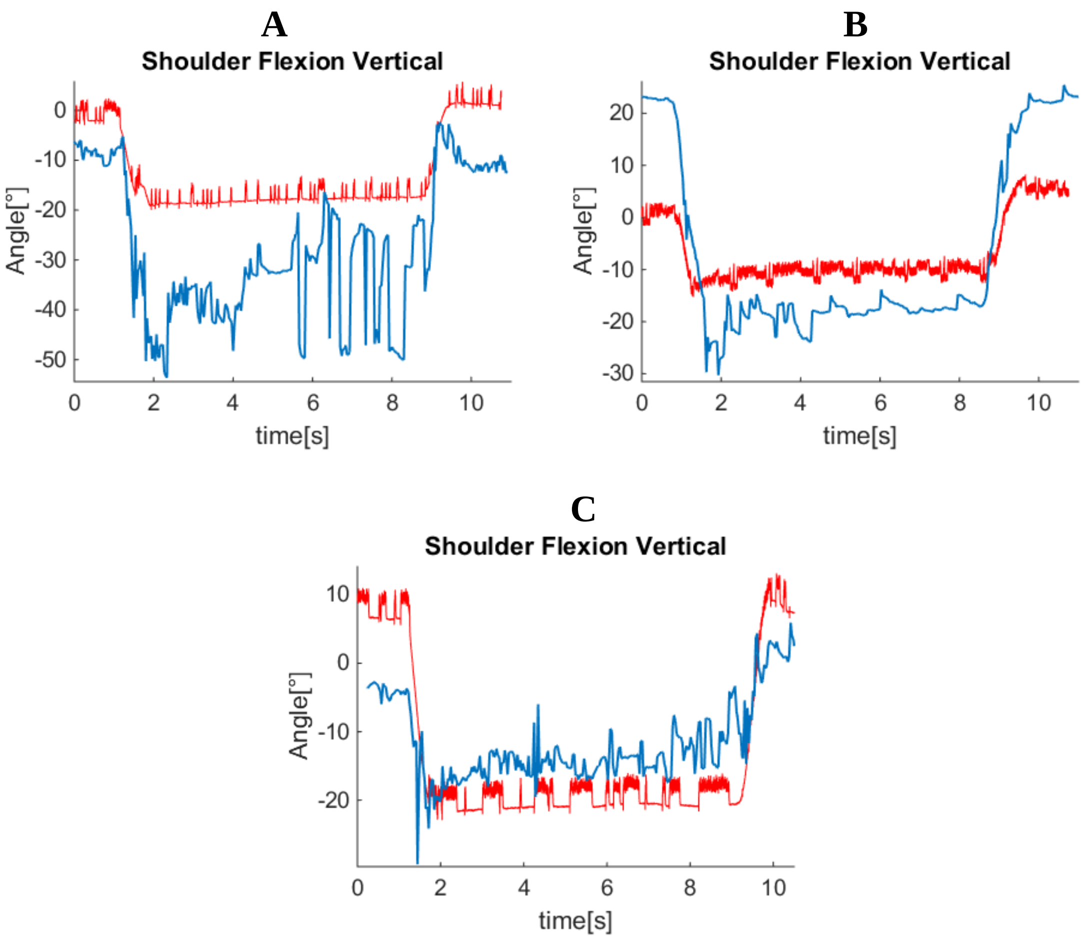



Figure 36: Shoulder vertical flexion from "full range of motions", arms forward, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

All three measurements (Figure 36) show very jumpy results, while the arms are stretched forward. The highest changes in the amplitude can be seen in the first measurement (Figure 36, A), followed by the third measurement (Figure 36, C), the second measurement (Figure 36, B) shows the least changes. The initial angle for the second measurement (Figure 36, B) is located near 20°, for the first measurement (Figure 36, A) near -10°, and for the third one (Figure 36, C) around 0°.

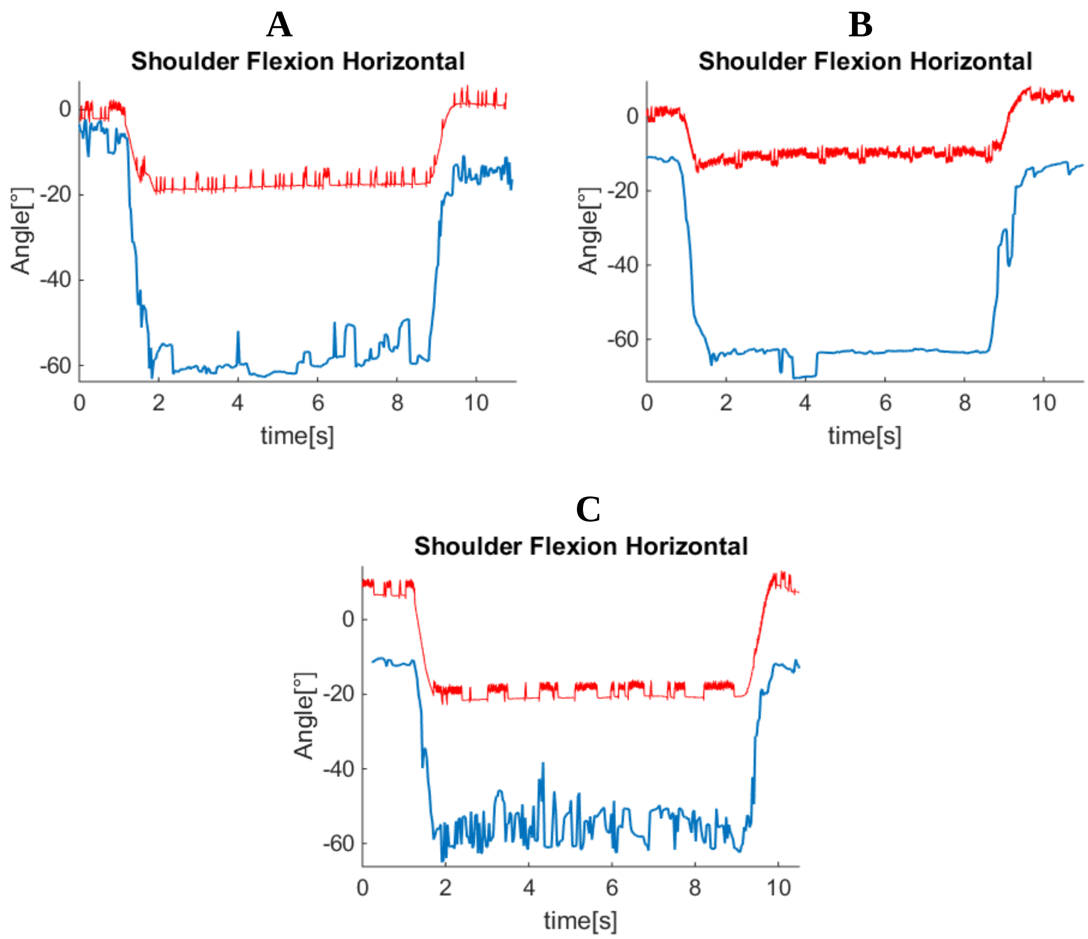



Figure 37: Shoulder horizontal flexion from "full range of motions", arms forward, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

The result of the third measurement (Figure 37, C) is very jumpy during the stretching, compared to the other two (Figure 37, A, B). Although the first measurement (Figure 37, A) contains some irregularities itself, too. The best result derives from the second measurement (Figure 37, B), containing only a few peaks and is otherwise constant. Within the first measurement (Figure 37, A) the angle at the initial hand position at the end of the measurement differs from the start of the measurement, while it remains quite constantly for the other two measurements (Figure 37, B, C). The maximum angle is nearly the same for all three measurements (Figure 37), only the starting angle is different. The first measurement (Figure 37, A) starts around 0°, the second one and the third one (Figure 37, B, C) starts approximately at -10°.

Detection of crossed arms at different locations was accomplished in the category “arm crossing”. Two tasks were performed: cross arms in front of the body and cross arms above the head.

## Arms crossed in front of the Body

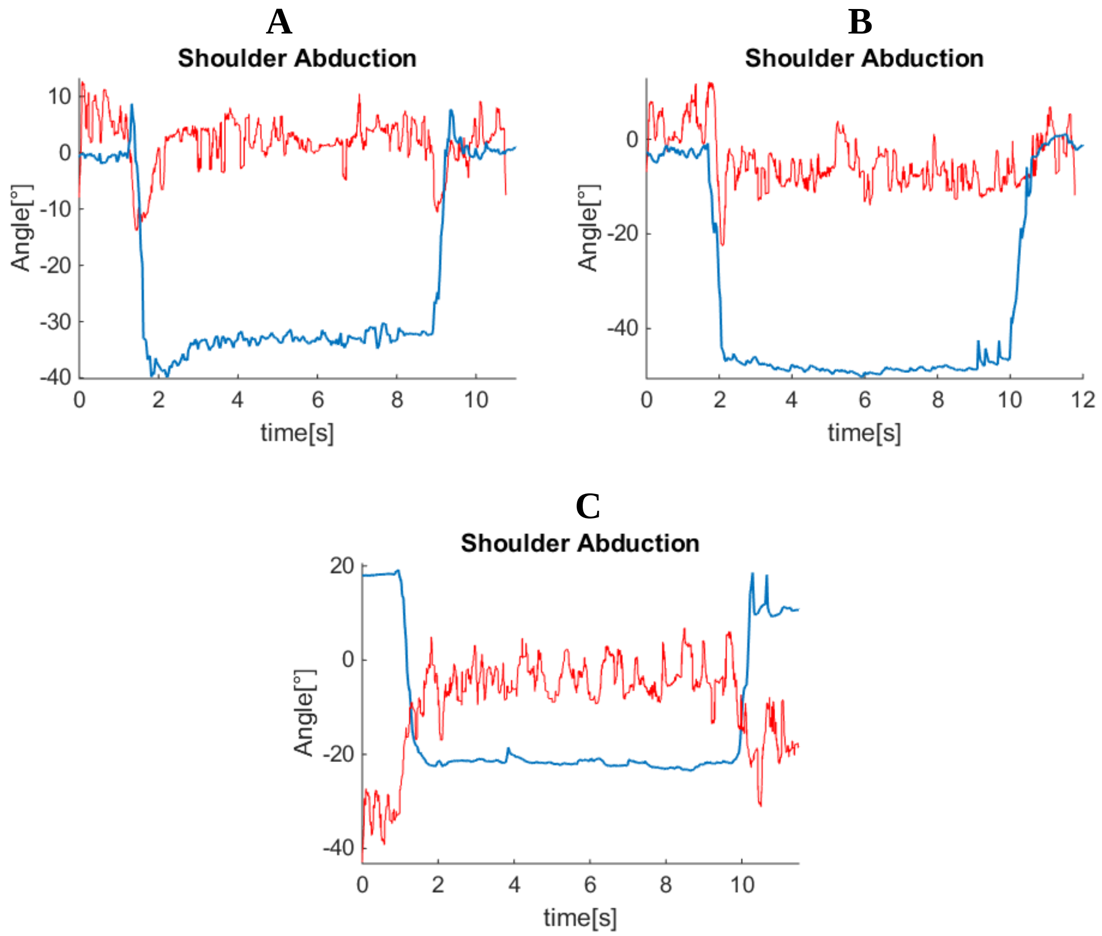



Figure 38: Shoulder abduction from “arm crossing”, arms crossed in front of the body, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

The second measurement (Figure 38, B) has the largest amplitude compared to the other two. However, the first & third measurement (Figure 38, A, C) have approximately the same amplitude and are only shifted by its starting value. Additionally, the results of the first measurement (Figure 38, A) are jumpy during the crossing, while the other ones (Figure 38, B, C) are nearly constant. During the transition from the initial hand position into the crossing and vice versa, positive peaks occur within the first measurement (Figure 38, A), and at the end of the third measurement (Figure 38, C), but not in the second one (Figure 38, B). The starting values for the first & second measurement (Figure 38, A, C) is nearly the same (~0°), and the third one (Figure 38, C) is around 20°.

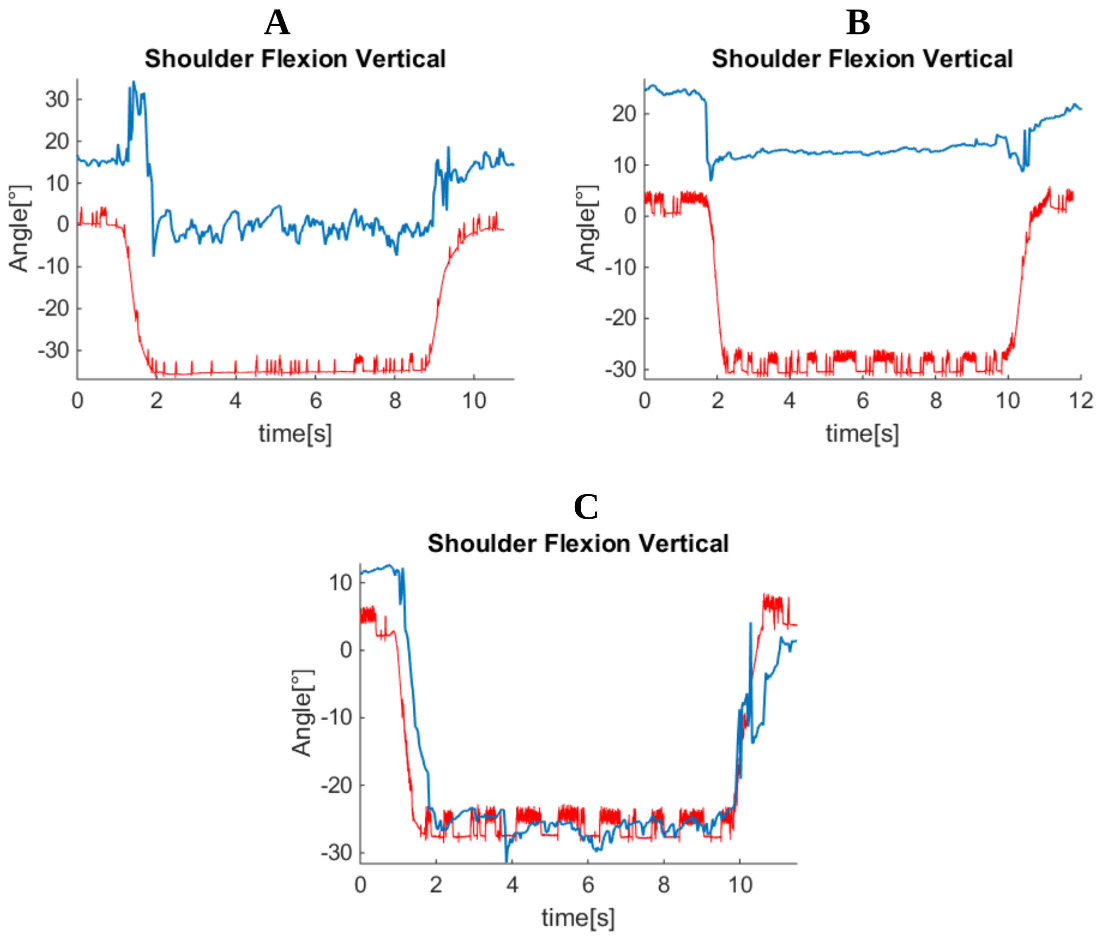



Figure 39: Shoulder vertical flexion from "arm crossing", arms crossed in front of the body, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

Jumpy data occur within the first & third measurement (Figure 39, A, C), while the second one (Figure 39, B) remains relatively constant during the crossing process. Also, the first measurement (Figure 39, A) consists of a positive peak during the transition of the initial hand position into the crossing, however, no peak occurs in the second & third measurement (Figure 39, B, C) at this point. A negative peak can be seen at the end of the data in the second & third measurement (Figure 39, B, C) at the transition of the crossing back to the initial hand state. The largest amplitude appears in the results of the third measurement (Figure 39, C), while the result of the other two measurements (Figure 39, A, B) is lower. Also, the starting point of each measurement is different, for the first measurement (Figure 39, A) the starting point is around 15°, for the second one (Figure 39, B) around 25° and the third one (Figure 39, C) around 10°.

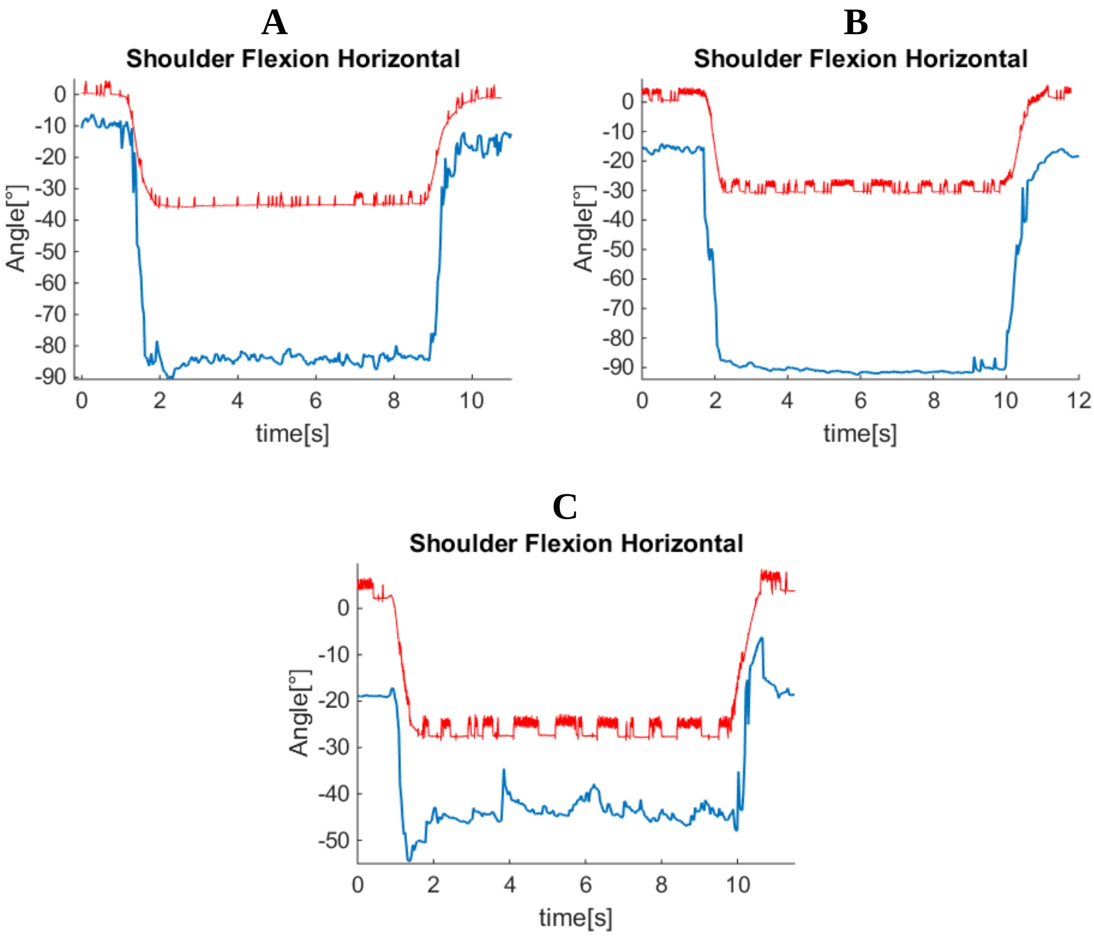



Figure 40: Shoulder horizontal flexion from "arm crossing", arms crossed in front of the body, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

The third measurement (Figure 40, C) contains jumpy and irregular values during the crossing, as well as the results from the first measurement (Figure 40, A), but not as large. In comparison the second measurement (Figure 40, B) shows a constant angle during the crossing. Within the initial hand position, some jumpy values occur in the graph of the first measurement (Figure 40, A), but not in the other ones (Figure 40, B, C). Instead, a positive peak is visible in the third measurement (Figure 40, C). The amplitudes of the first & second measurement (Figure 40, A, B) are nearly at the same level, while the third one (Figure 40, C) is smaller. Also, the starting values of all three measurements are different, the first one (Figure 40, A) is around -10°, the second one (Figure 40, B) is around -15°, and the last one (Figure 40, C) is around -20°.

### Arms crossed above the Head

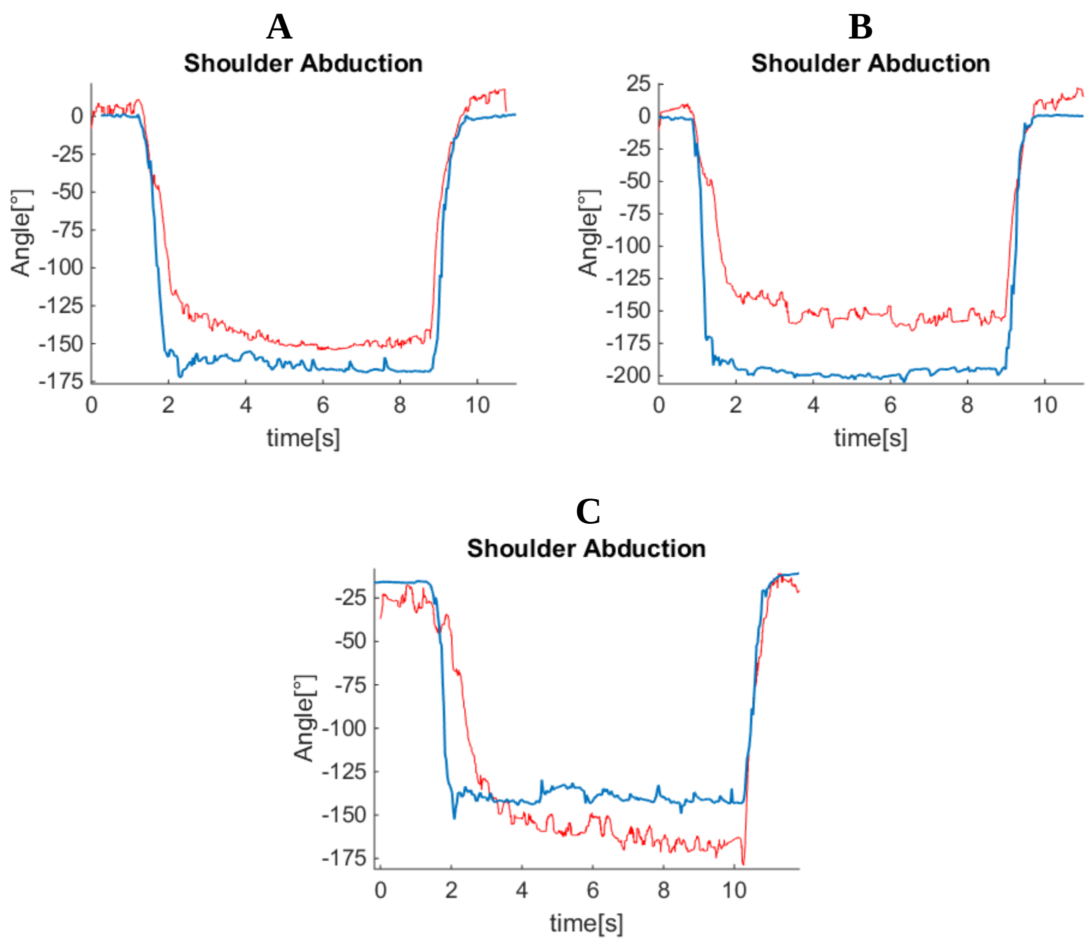



Figure 41: Shoulder abduction from "arm crossing", arms crossed above the head, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

All data show a certain jumpiness, the third measurement (Figure 41, C) contains the highest jumpiness, and the second measurement (Figure 41, B) contains the lowest jumpiness, while the arms are crossed. The highest amplitude was found within the second measurement (Figure 41, B), and the lowest amplitude was found in the third measurement (Figure 41, C). The starting angles for the first & second measurement

(Figure 41, A, B) is similar to each other (~0°), however, the starting angle for the third measurement (Figure 41, C) is near -25°.

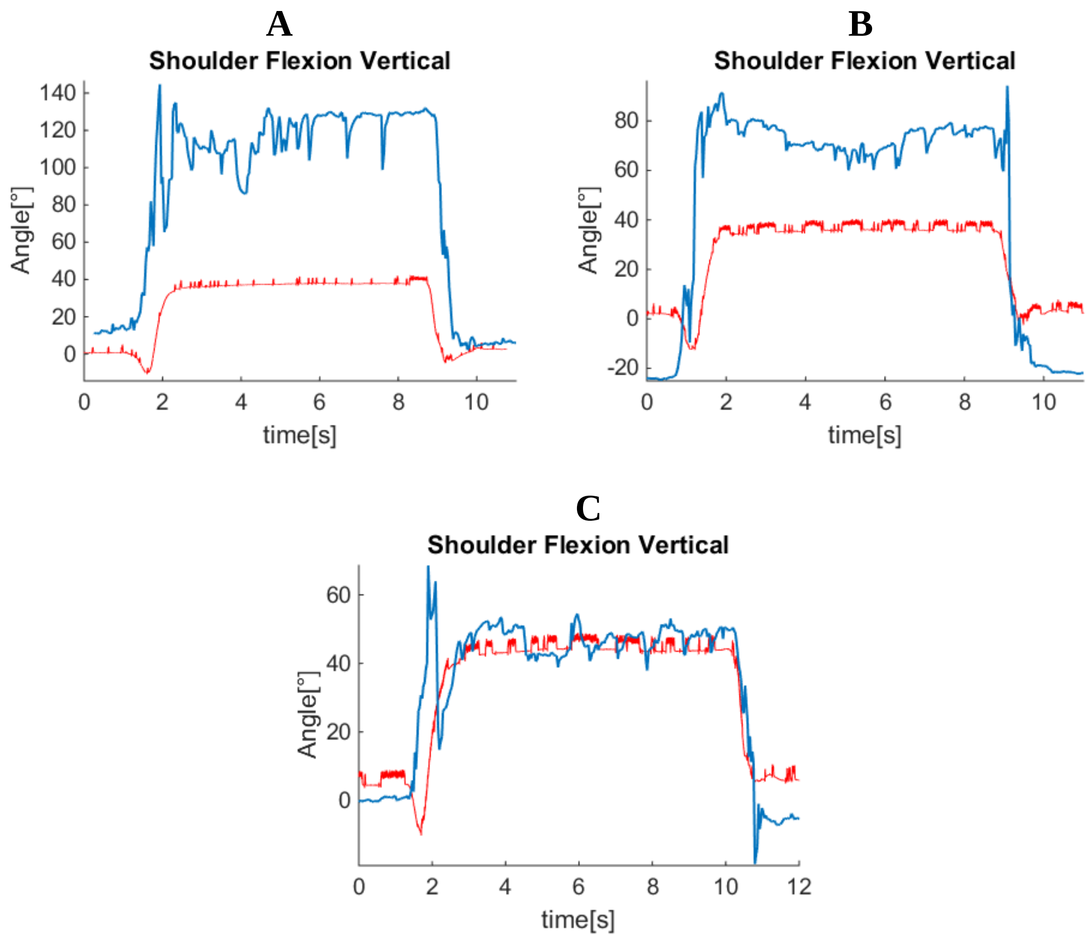



Figure 42: Shoulder vertical flexion from “arm crossing”, arms crossed above the head, Kinect$^{TM}$ in front with desk (A), desk turned 30°-45° (B), Kinect$^{TM}$ in front no desk (C)

All three measurements (Figure 42) show very jumpy results, while the arms are crossed. The highest changes in the amplitude can be seen in the first measurement (Figure 42, A), followed by the second measurement (Figure 42, B), and the third measurement (Figure 42, C) shows the least changes. In the third measurement (Figure 42, C) a positive peak occur at the transition of the initial hand state into the crossing, similar to the first measurement (Figure 42, A). Additionally, a negative peak occurs within the third measurement (Figure 42, C) during the transition of the crossing back into the initial hand state. The initial angle for the second measurement (Figure 42, B) is located near -20°, for the first measurement (Figure 42, A) near 10°, and for the third one (Figure 42, C) around 0°.

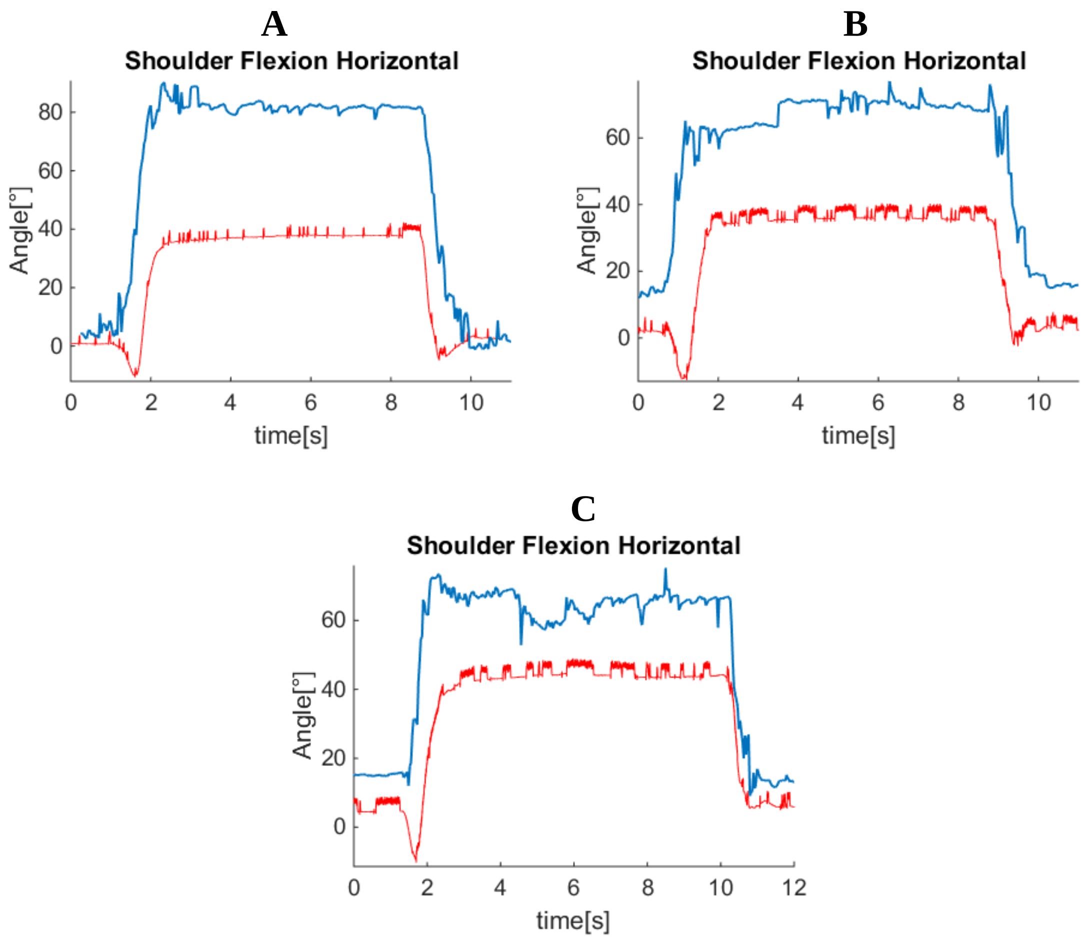



Figure 43: Shoulder horizontal flexion from "arm crossing", arms crossed above the head, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

Fundamentally, the results of all measurements (Figure 43) show jumpy and irregular values during the crossing, in which the largest changes arise in the third measurement (Figure 43, C), and the fewest ones in the first measurement (Figure 43, A). Also, some changes occur at the starting & ending position of the first & second measurement (Figure 43, A, B), while in the third measurement (Figure 43, C) the changes only appear at the ending position. The first measurement (Figure 43, A) has the largest amplitude compared to the other ones. However, the second & third measurement (Figure 43, B, C) have approximately the same amplitude. The second & third measurement (Figure 43, B, C) have a similar starting point (~15°), while the starting point of the first is around 0°.

Within “grasping an object” an object (*bottle, cup*) was moved from a given position to another position (*different high levels*). In total four tasks were performed.

Grasping an Object (Bottle, no height)

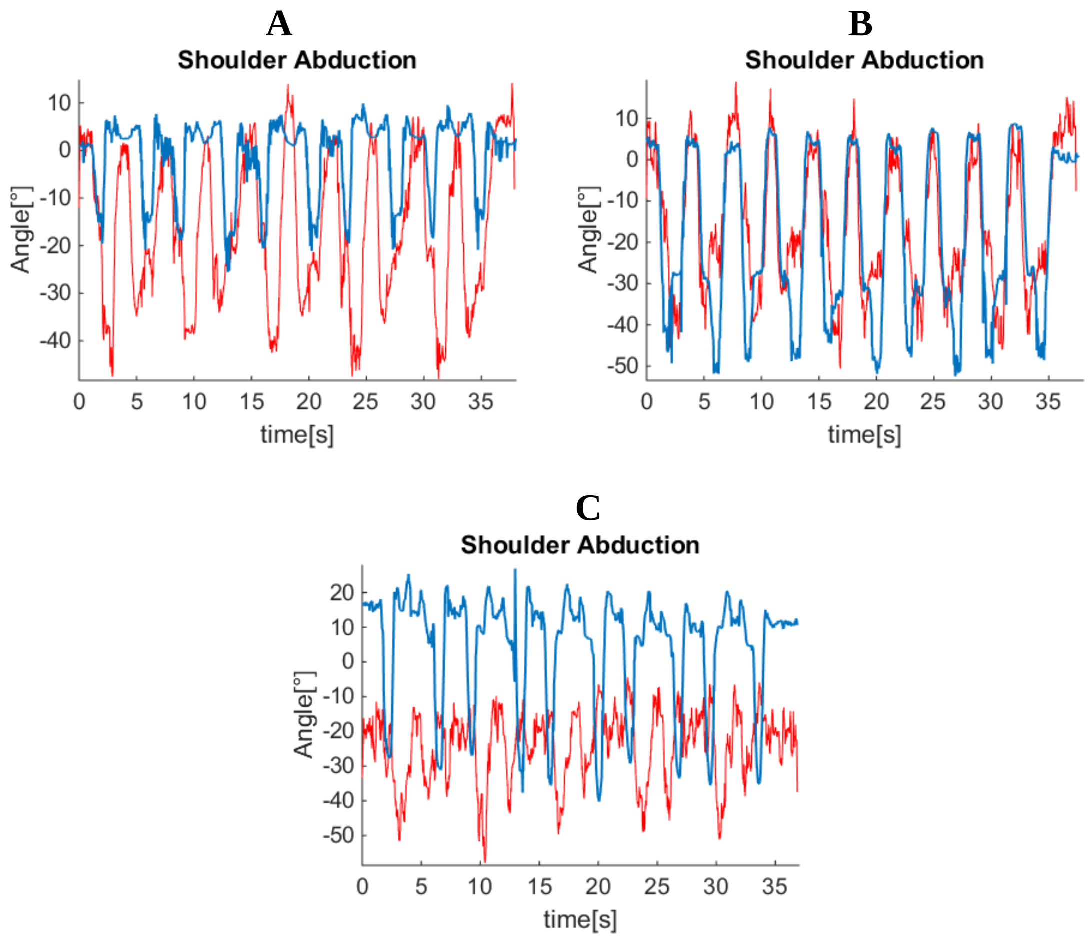



Figure 44: Shoulder abduction from “grasping an object”, grasp bottle no height, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

Within the results of the second measurement (Figure 44, B) it can be seen, that all peaks were correctly detected, while in the results of the first & third measurement (Figure 44, A, C) some peaks are missing. Furthermore, irregularities occur in the first & third measurement (Figure 44, A, C), but not in the second measurement (Figure 44, B). Additionally, the range of values is large for the second measurement (Figure 44, B) (-

55° up to 5°), a little bit lower for third measurement (Figure 44, C) (-35° up to 20°), and is smallest for the first one (Figure 44, A) (-20° up to 5°). Also, the starting point for the first & second measurement (Figure 44, A, B) is near 0°, while the starting point for the third measurement (Figure 44, C) is near 20°.

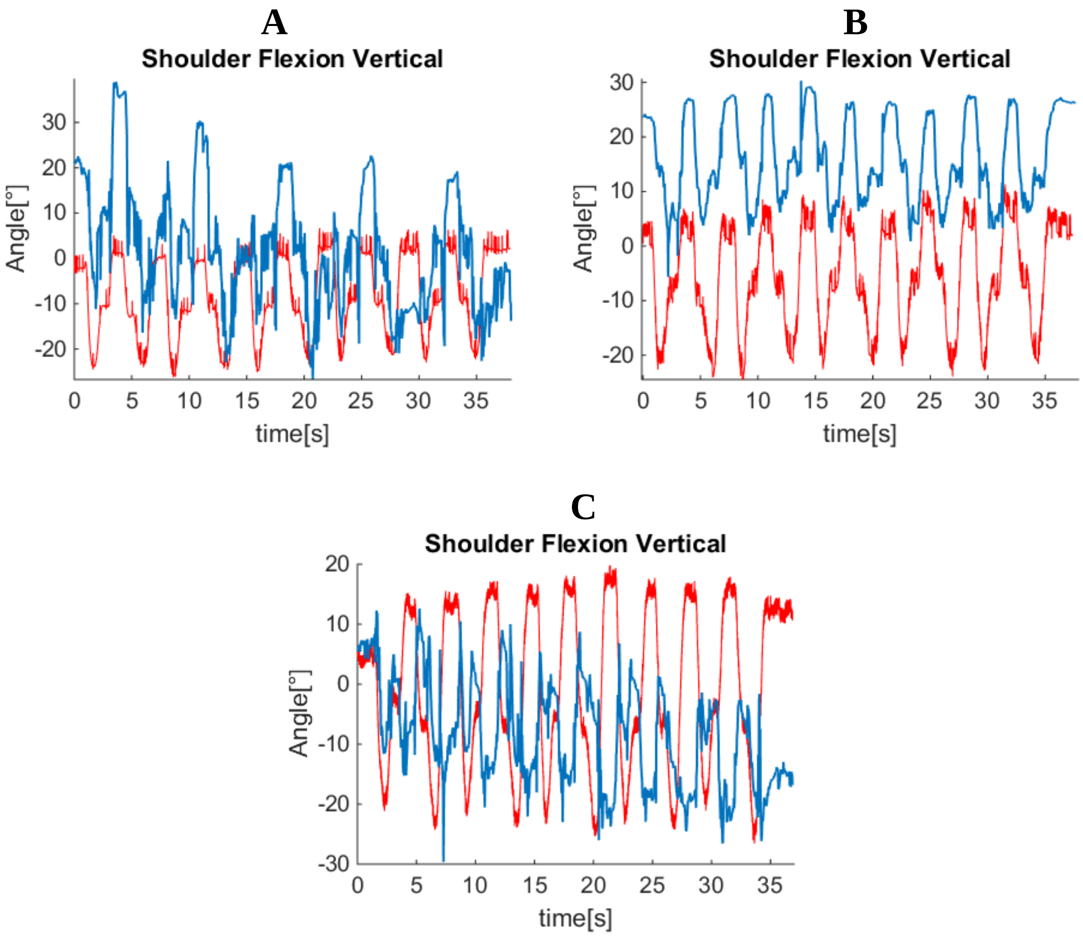



Figure 45: Shoulder vertical flexion from "grasping an object", grasp bottle no height, Kinect$^{TM}$ in front with desk (A), desk turned 30°-45° (B), Kinect$^{TM}$ in front no desk (C)

Here, the first & third measurement (Figure 64, A, C) show a decreasing amplitude, during the performance of the motion, while the second measurement (Figure 64, B) has a constant amplitude. The first & third measurement (Figure 64, A, C) consist of jumpy values, compared to the second one (Figure 64, B). Further, within the results of the second measurement (Figure 64, B) all peaks were correctly detected, while within the results for the first measurement (Figure 64, A) peaks were missing, and a statement about the missing peaks within the results of the third measurement (Figure 64, C) cannot be given. For all three measurements (Figure 64) the starting value differs from each other, first measurement has ~20°, second one has ~25° and the third one has ~5°.

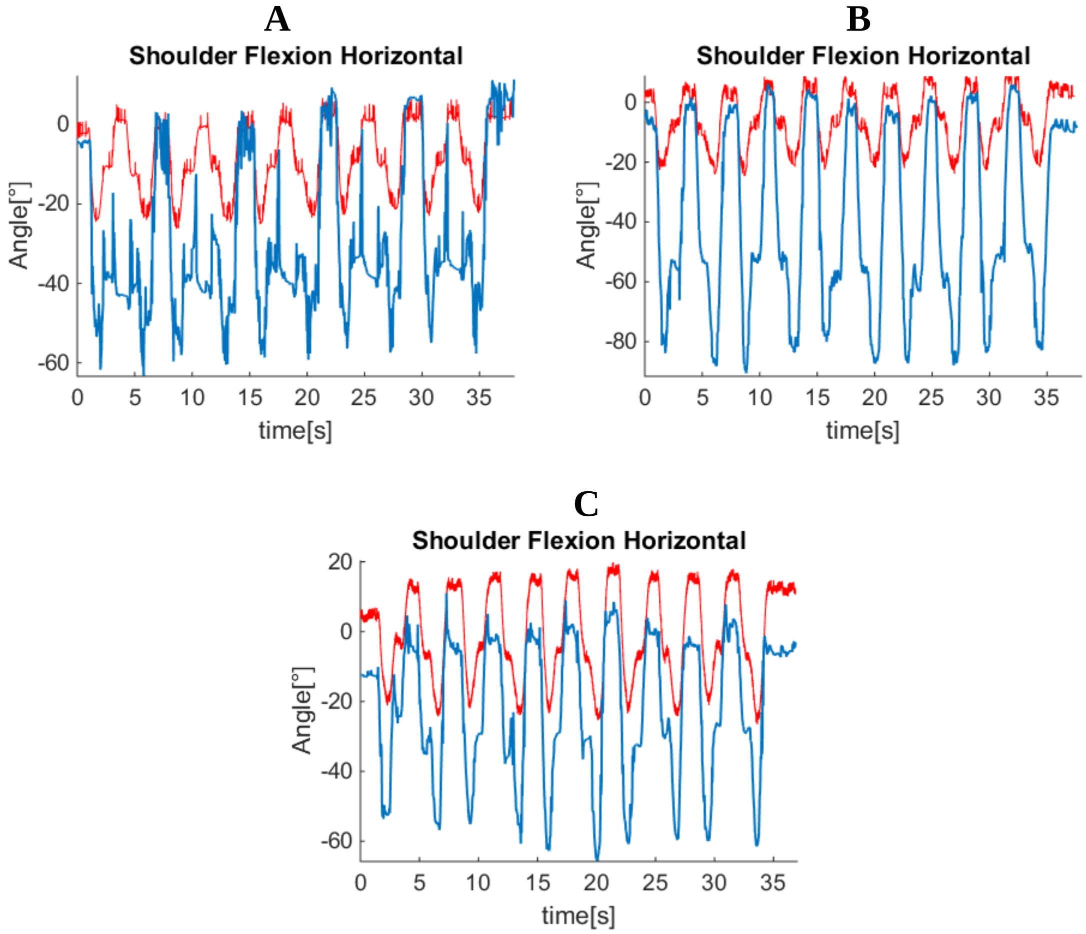



Figure 46: Shoulder horizontal flexion from "grasping an object", grasp bottle no height, Kinect$^{TM}$ in front with desk (A), desk turned 30°-45° (B), Kinect$^{TM}$ in front no desk (C)

All peaks of the second & third measurement (Figure 46,B, C) were detected correctly, although, not all peaks were detected in the first measurement (Figure 46, A). Additionally, the results of the first measurement (Figure 46, A) consist of jumpy values, compared to the other two (Figure 46, B, C). The range of values in the first & third measurement (Figure 46, A, C) is similar to each other (-60° up to 0°), while the range of values for the second measurement (Figure 46, B) is larger (-90° up to 0°). Also, the starting values for the first & second measurement are alike (~0°), and differ from the third measurement (Figure 46, C) (~-10°).

Grasping an Object (Bottle, height: 15cm)

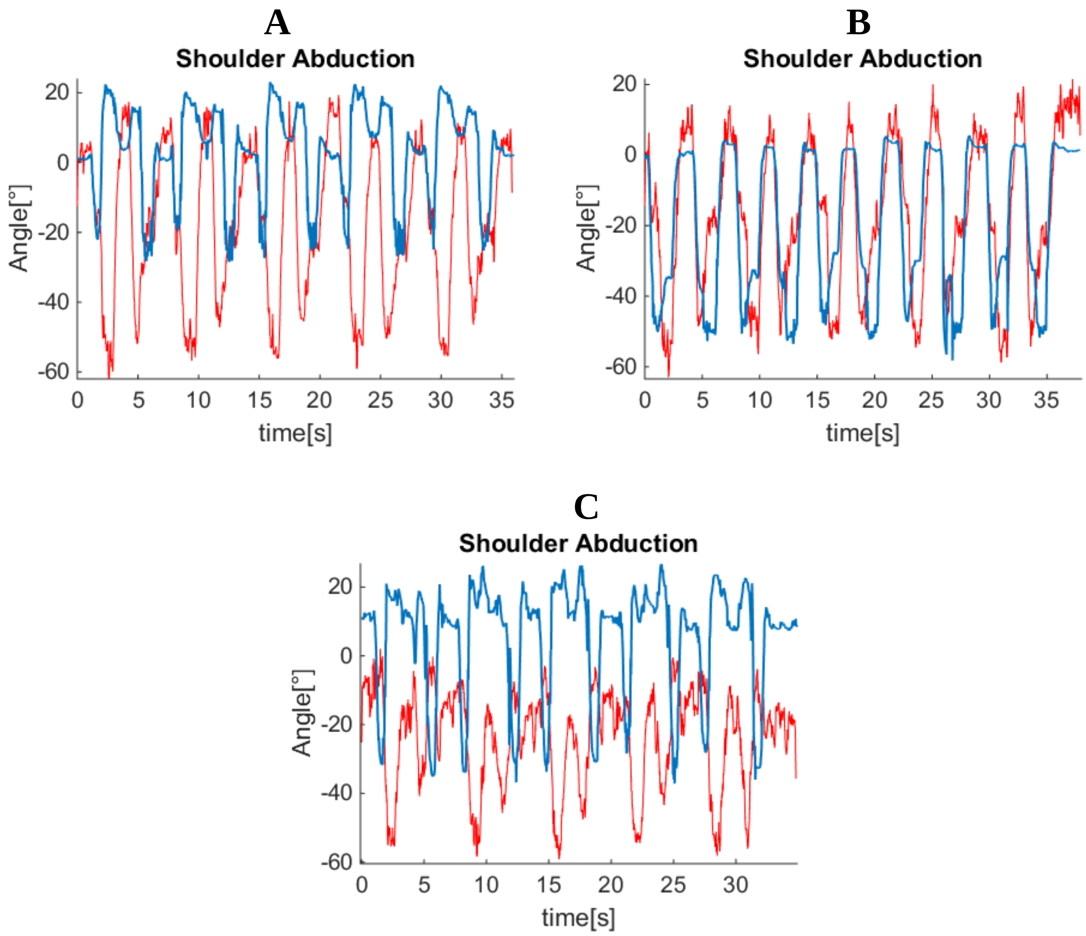



Figure 47: Shoulder abduction from "grasping an object", grasp bottle 15cm height, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

In the results of the first & third measurement (Figure 47. A, C) missing peaks can be found, while in the results of the second measurement (Figure 47, B) no peaks are missing. The range of values is different for all three measurements (Figure 47). Starting with the first measurement (Figure 47, A) the range is around -20° up to 20°, for the second measurement (Figure 47, B) approximately -50° up to 0°, and for the third measurement (Figure 47, C) -30° up to 20°. The starting angle for the first & second measurement (Figure 47, A, B) is nearly the same (~0°), while the starting value for the third measurement (Figure 47, C) differs (~15°).

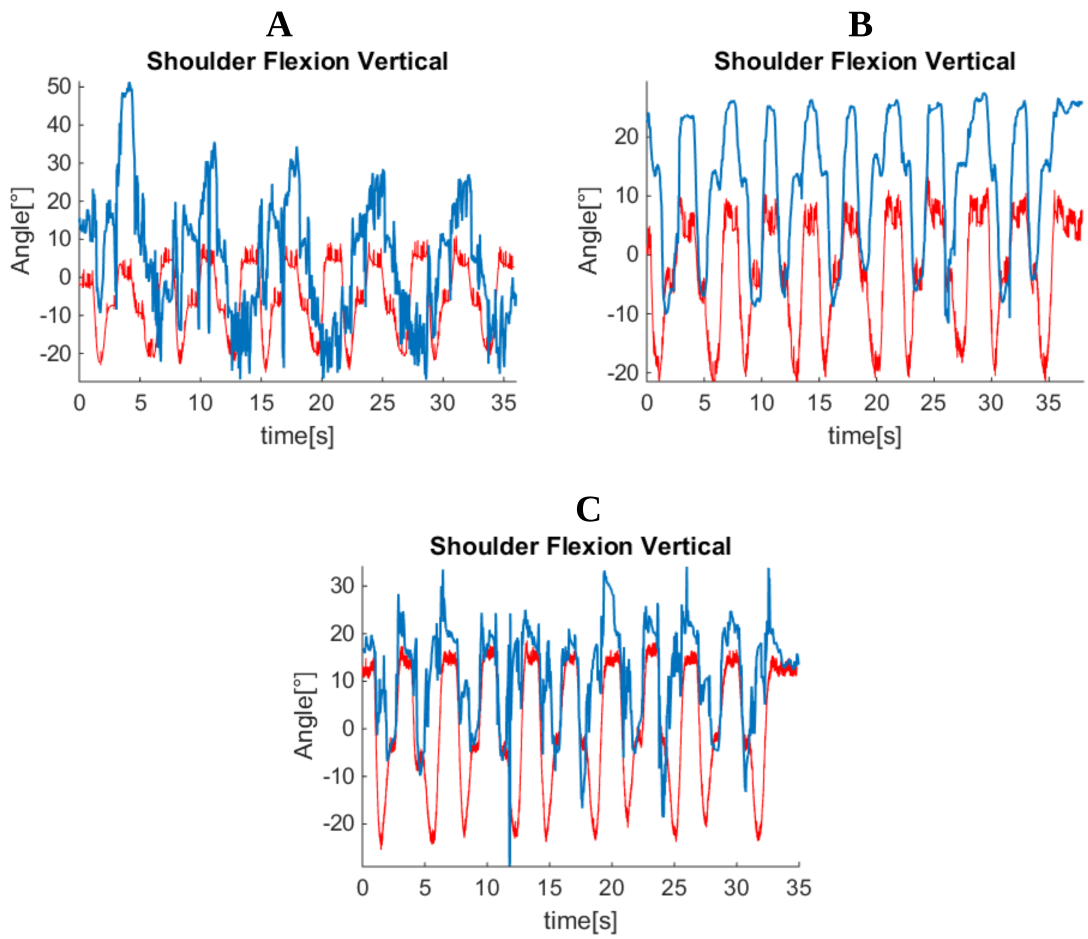



Figure 48: Shoulder vertical flexion from "grasping an object", grasp bottle 15cm height, Kinect$^{TM}$ in front with desk (A), desk turned 30°-45° (B), Kinect$^{TM}$ in front no desk (C)

Within the first measurement (Figure 48, A), jumpy & irregular values are noticeable, and also consist of a decreasing amplitude over time. In comparison to that (Figure 48, A), the results of the third measurement (Figure 48, C) are jumpy and irregular as well, but don't decrease over time, while the results of the second measurement (Figure 48, B) remain constant and have none of these properties. Additionally, all peaks in the second & third measurement (Figure 48, B, C) were detected correctly, while missing peaks occur in the first measurement (Figure 48, A). The range of values is large for the first measurement (Figure 48, A) (-25° up to 50°), a little bit lower for the third measurement (Figure 48, C) (-20° up to 30°), and is smallest for the second one (Figure 48, B) (-10° up to 25°). The second measurement (Figure 48, B) starts around 25°, the first and the third one (Figure 48, A, C) starts approximately at 15°.

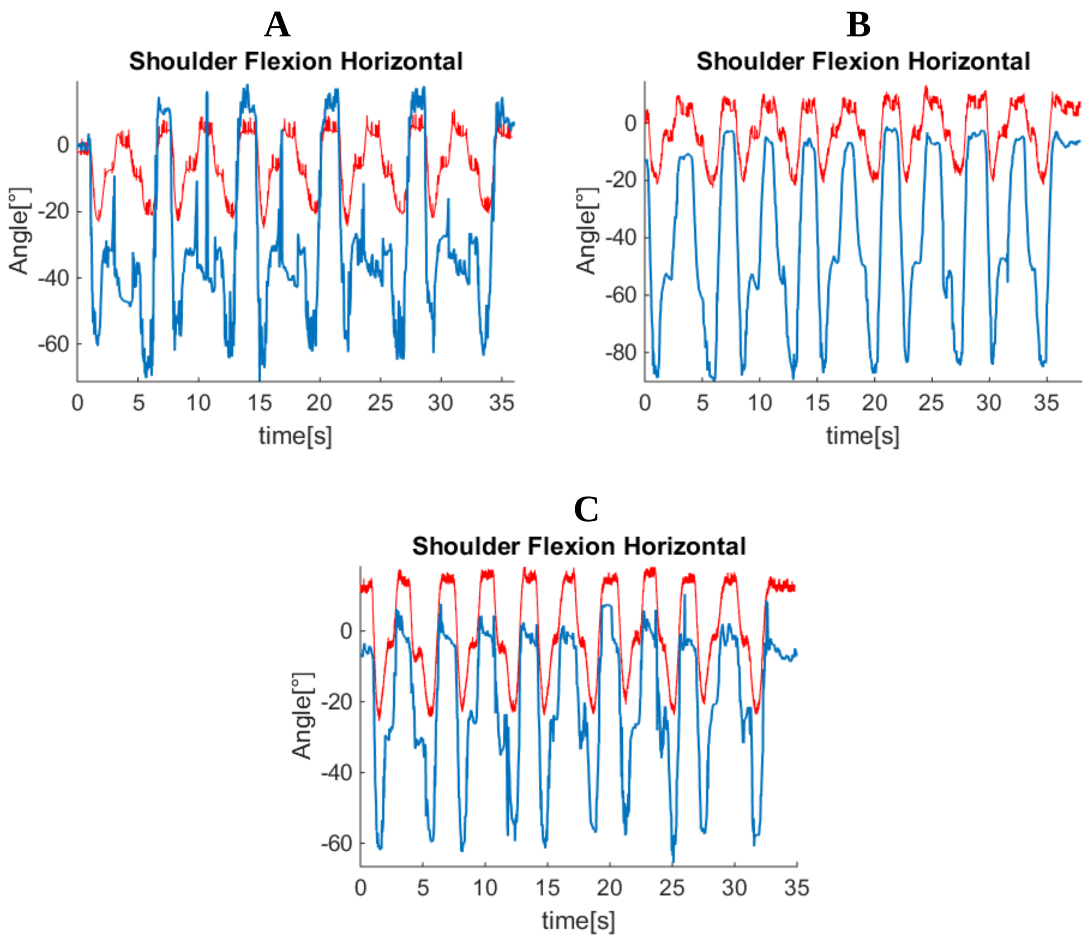



Figure 49: Shoulder horizontal flexion from “grasping an object”, grasp bottle 15cm height, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

When comparing all three measurements (Figure 49) it can be ascertained, that not all peaks were detected correctly within the first measurement (Figure 49, A). Further, jumpy and irregular values occur within the results of the first measurement (Figure 49, A), and slightly within the results of the third measurement (Figure 49, C), while the second measurement (Figure 49, B) remains constant. Differences within the range of values between the measurement are noticeable,  the largest range (-90° up to -0°) arises within the second measurement (Figure 49, B), the lowest (-60° up to 0°) within the third measurement (Figure 49, C), and the first measurement (Figure 49, A) in between (-70° up to 15°). The starting value for the first measurement (Figure 49, A) is around 0°, for the second measurement (Figure 49, B) around -15°, and for the third measurement (Figure 49, C) around -5°.

### Grasping an Object (Cup, no height)

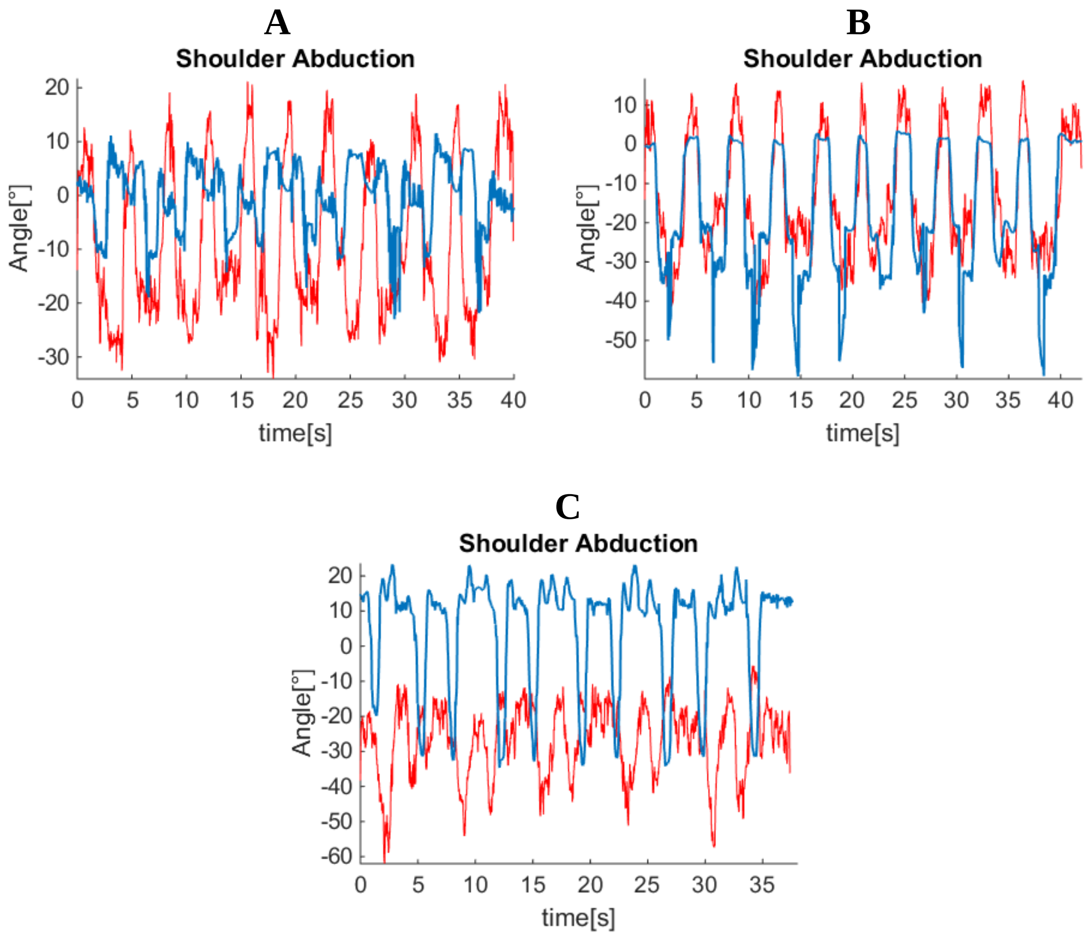



Figure 50: Shoulder abduction from "grasping an object", grasp cup no height, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

Within the results of the first & third measurement (Figure 50, A, C) it can be seen, that not all peaks were correctly detected, compared to the results of the second measurement (Figure 50, B). Further, the first measurement (Figure 50, A) consists of jumpy and irregular values, while the other two (Figure 50, B, C) do not. Starting with the first measurement (Figure 50, A) the range is around -15° up to 10°, for the second measurement (Figure 50, B) approximately -60° up to 0°, and for the third measurement (Figure 50, C) -30° up to 20°. The starting angle for the first & second measurement (Figure 50, A, B) is nearly the same (~0°), whereas the starting value for the third measurement (Figure 50, C) is different (~15°).

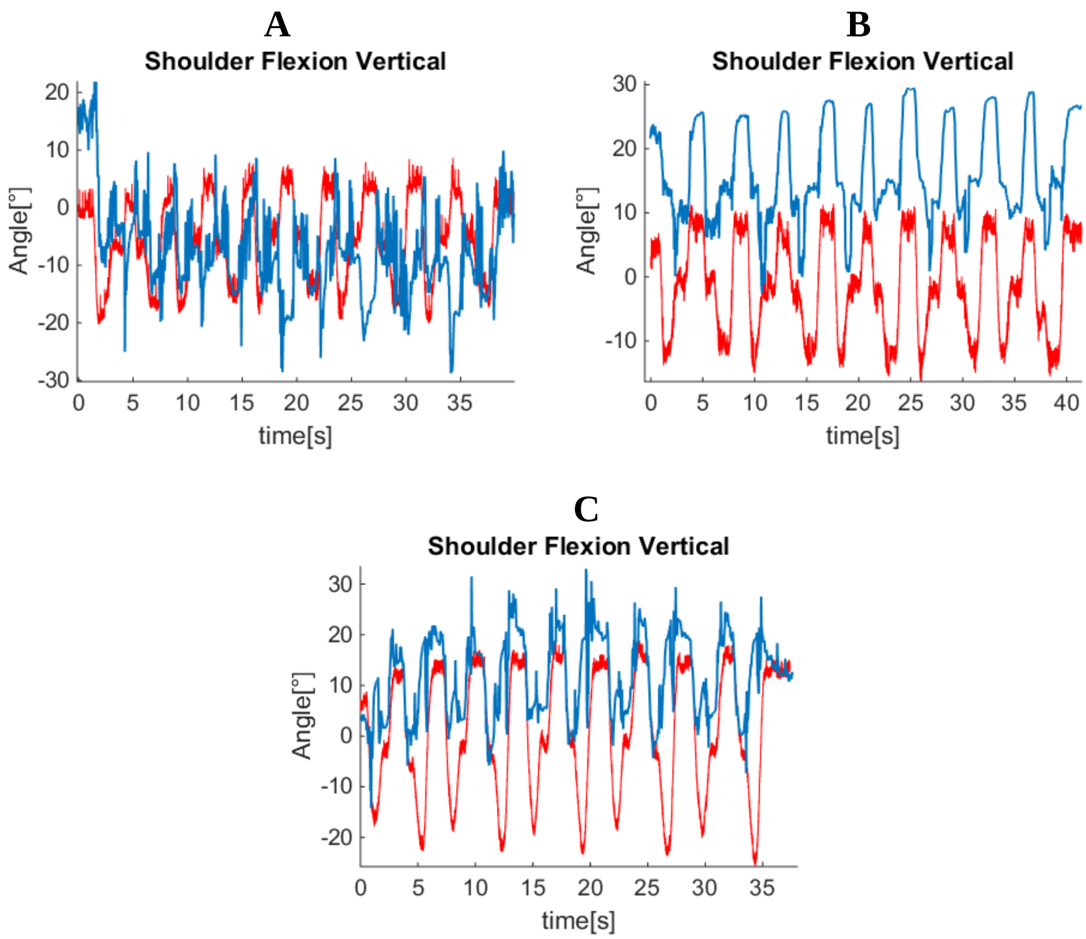



Figure 51: Shoulder vertical flexion from "grasping an object", grasp cup no height, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

A lot of jumpy and irregular values can be ascertained within the first & third measurement (Figure 51, A, C), while only a few occur within the second one (Figure 51, B). Further, not all peaks were detected correctly in the first measurement (Figure 51, A), in comparison to the second & third one (Figure 51, B, C). Additionally, the range of values is large for the third measurement (Figure 51, C) (-5° up to 30°), a little bit lower for the second measurement (Figure 51, B) (0° up to 30°), and are smallest for the first one (Figure 51, A) (-20° up to 5°). The starting value for the third measurement (Figure 51, C) is near 0°, for the first one (Figure 51, A) near 15° and for the second one (Figure 51, B) near 20°.

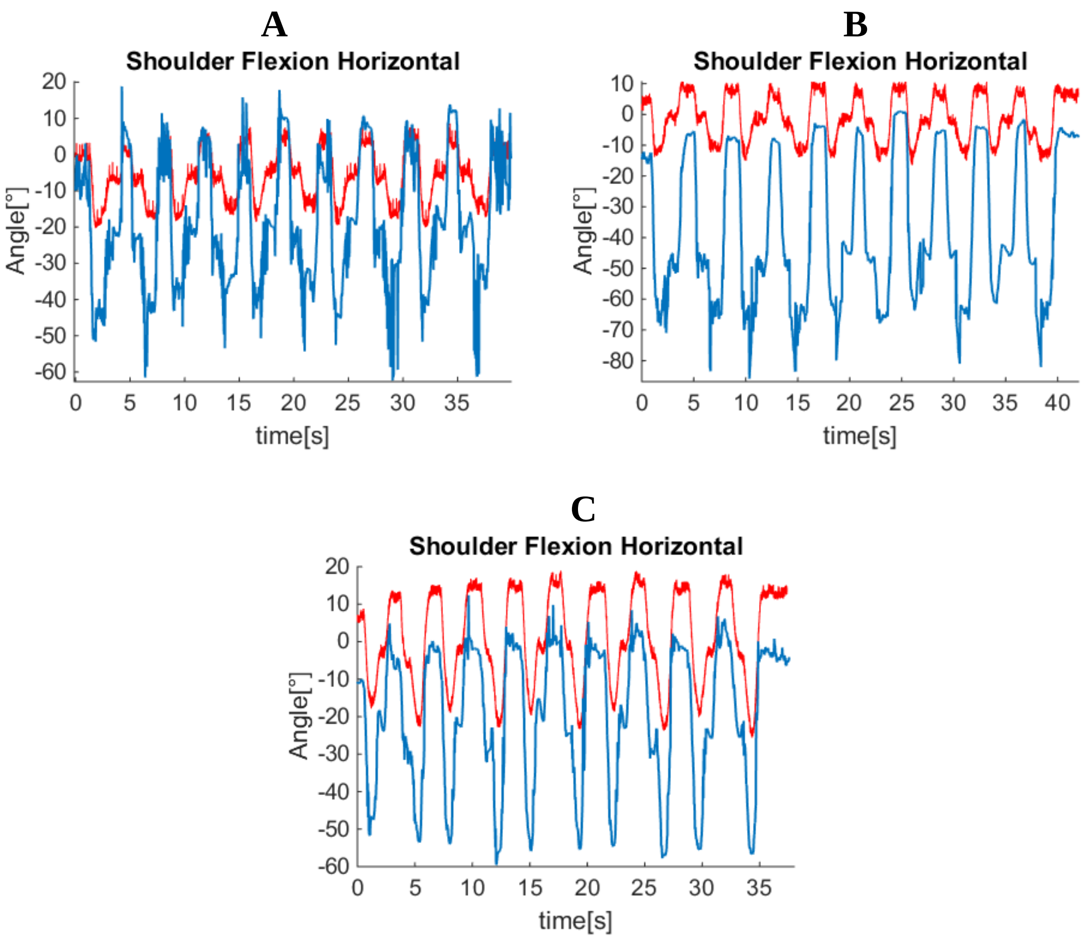



Figure 52: Shoulder horizontal flexion from "grasping an object", grasp cup no height, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

The results of the first measurement (Figure 52, A) are very jumpy during the grasping, compared to the other two (Figure 52, B, C). However, the second & third measurement are also a bit jumpy, but not as much as the first one (Figure 52, A). The range of values of the first measurement (Figure 52, A) (-60° up to 10°) is similar the second measurement (Figure 52, B) (-80° up to -10°), while the third measurement (Figure 52, C) is slightly smaller (-60° up to 0°). The starting angle for the second & third measurement (Figure 52, B, C) is nearly the same (~-10°), whereas the starting value for the first measurement (Figure 52, A) is different (~-5°).

Grasping an Object (Cup, height: 15cm)

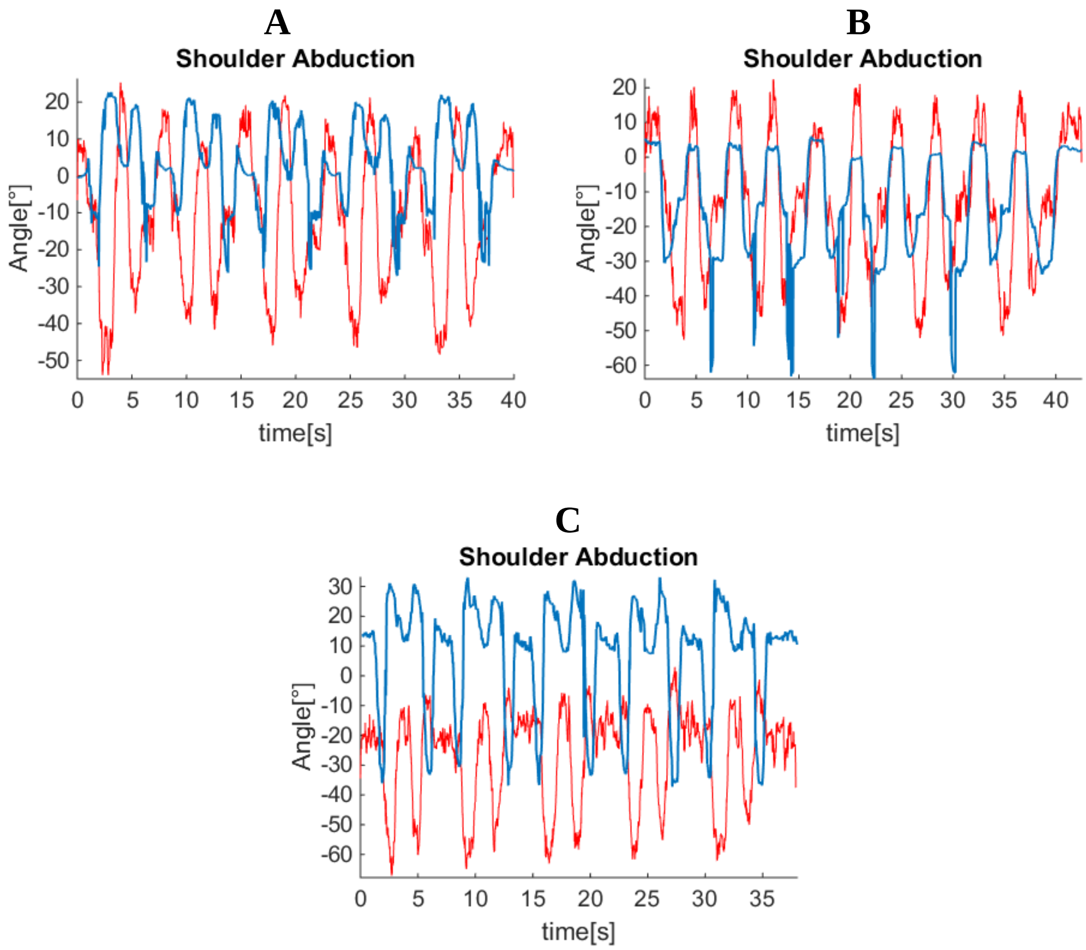



Figure 53: Shoulder abduction from "grasping an object", grasp cup 15cm height, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

Within the results of the second measurement (Figure 53, B) it can be seen, that all peaks were correctly detected, while in the results of the first & third measurement (Figure 53, A, C) some peaks are missing. Furthermore, negative spikes occur in the first & second measurement (Figure 53, A, B), but not in the third one (Figure 53, C). The starting point of each measurement is different, for the first measurement (Figure 53, A) the starting point is around 0°, for the second one (Figure 53, B) around 5° and for the third one (Figure 53, C) around 15°. Also the range of values varied, the largest range (-60° up to 5°) is found in the second measurement (Figure 53, B), and the smallest one (-20° up to 20°) in the first measurement (Figure 53, A). In between lays the range (-30° up to 30°) of the third measurement (Figure 53, C).

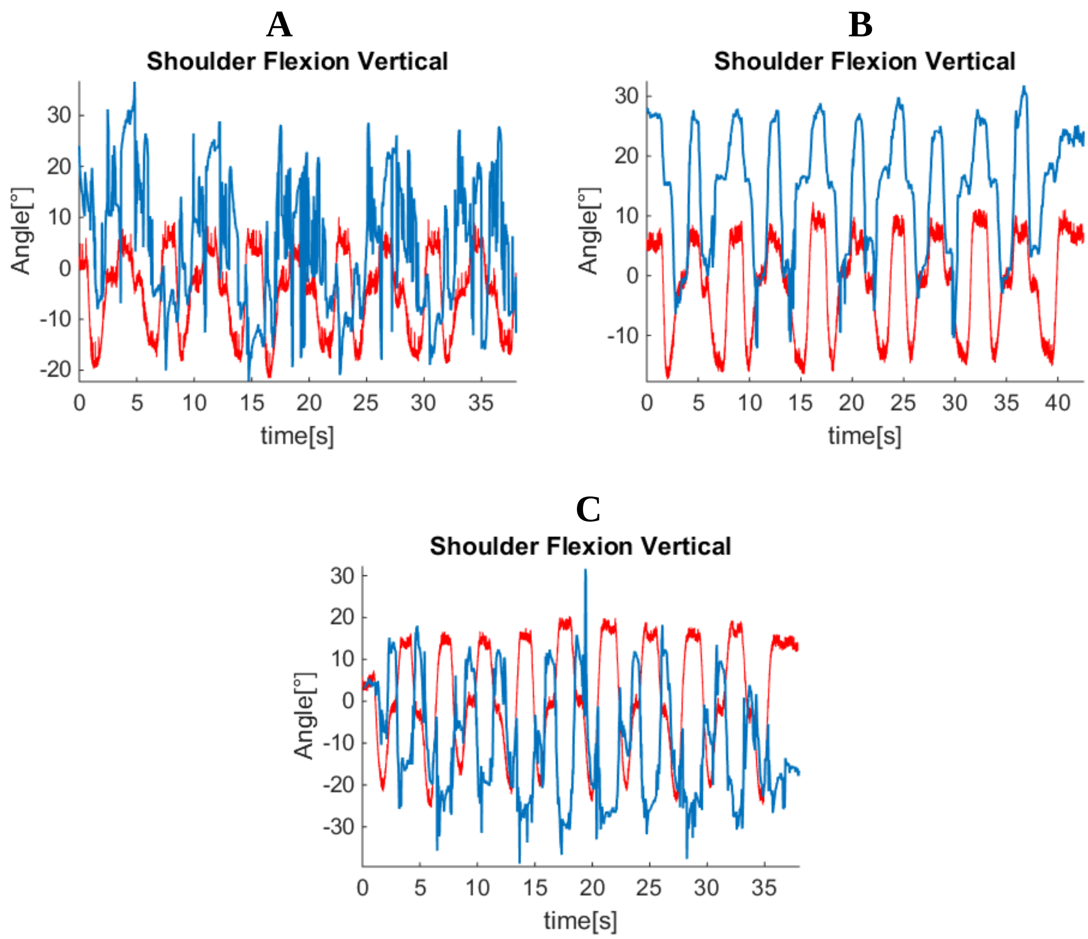



Figure 54: Shoulder vertical flexion from "grasping an object", grasp cup 15cm height, Kinect$^{TM}$ in front with desk (A), desk turned 30°-45° (B), Kinect$^{TM}$ in front no desk (C)

Very jumpy and irregular values occur within the results of the first measurement (Figure 54, A), and slightly within the results of the third measurement (Figure 54, C), while the second one (Figure 54, B) remains constant. It can also be noticed, that not all peaks were detected in the first & third measurement (Figure 54, A, C), compared to the second measurement (Figure 54, B). Further, the starting values of all three measurements are different, the first one (Figure 54, A) is around 20°, the second one (Figure 54, B) is around 30°, and the last one (Figure 54, C) is around 5°. The largest range of values is shown in the first measurement (Figure 54, A) (-20° up to 30°), followed by the third measurement (Figure 54, C) (-30° up to 15°) and the smallest range is located in the second measurement (Figure 54, B) (-5° up to 30°).

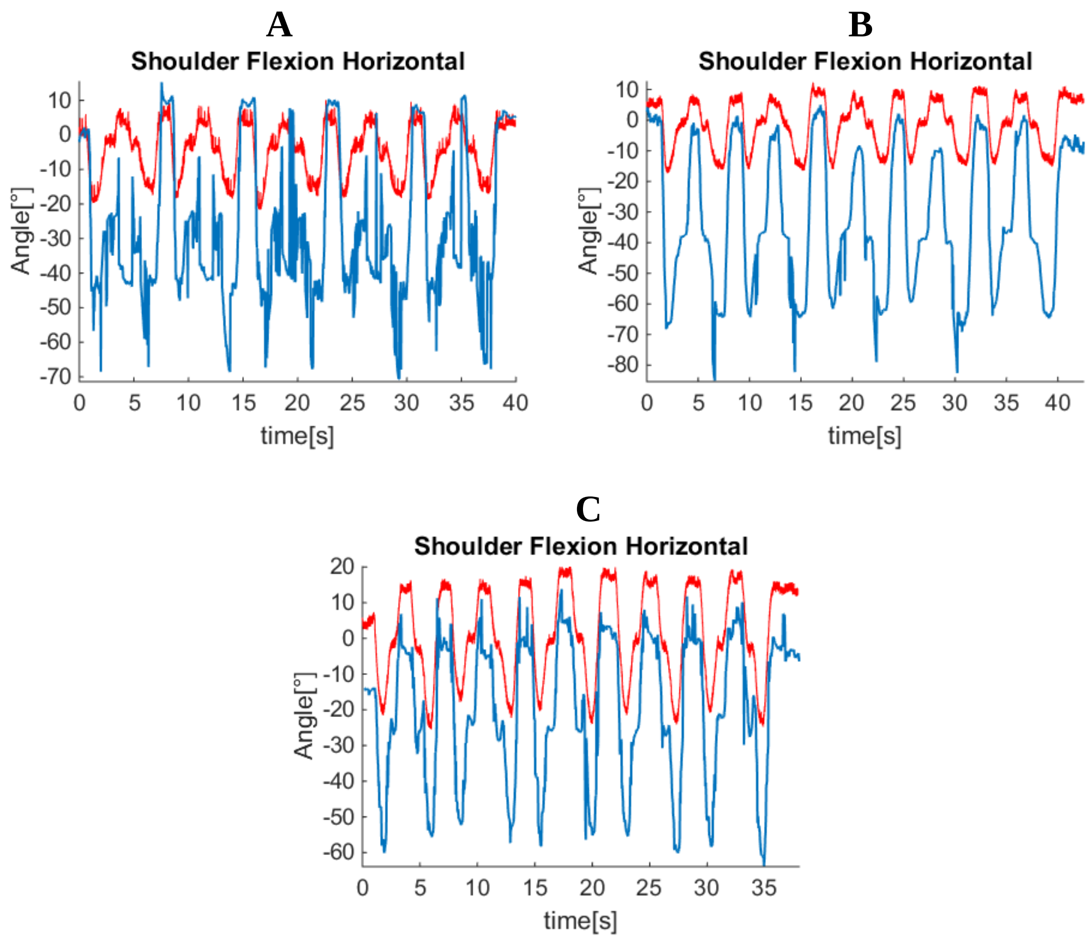



Figure 55: Shoulder horizontal flexion from "grasping an object", grasp cup 15cm height, Kinect™ in front with desk (A), desk turned 30°-45° (B), Kinect™ in front no desk (C)

In the results of the first measurement (Figure 55, A) missing peaks can be recognized, while in the results of the second & third measurement (Figure 55, B, C) no peaks are missing. Further, within the data of the first measurement (Figure 55, A) many jumpy and irregular values occur, although only a few occur in the second & third measurement (Figure 55, B, C). The range of the values of the first (-70° up to 10°) & second (-85° up to 0°) measurement (Figure 55, A, B) is similar to each other, while the range (-60° up to 10°) of the third measurement (Figure 55, C) is slightly different. Also, the starting values for the first & second measurement (Figure 55, A, B) are alike (~0°), and differ from the third measurement (Figure 55, C) (~-15°).

## RESULTS OBTAINED FROM THE ELBOW

To ensure a good readability not all obtained results of the elbow will be explained and graphically represented. Instead, one chosen example of the results will be described.

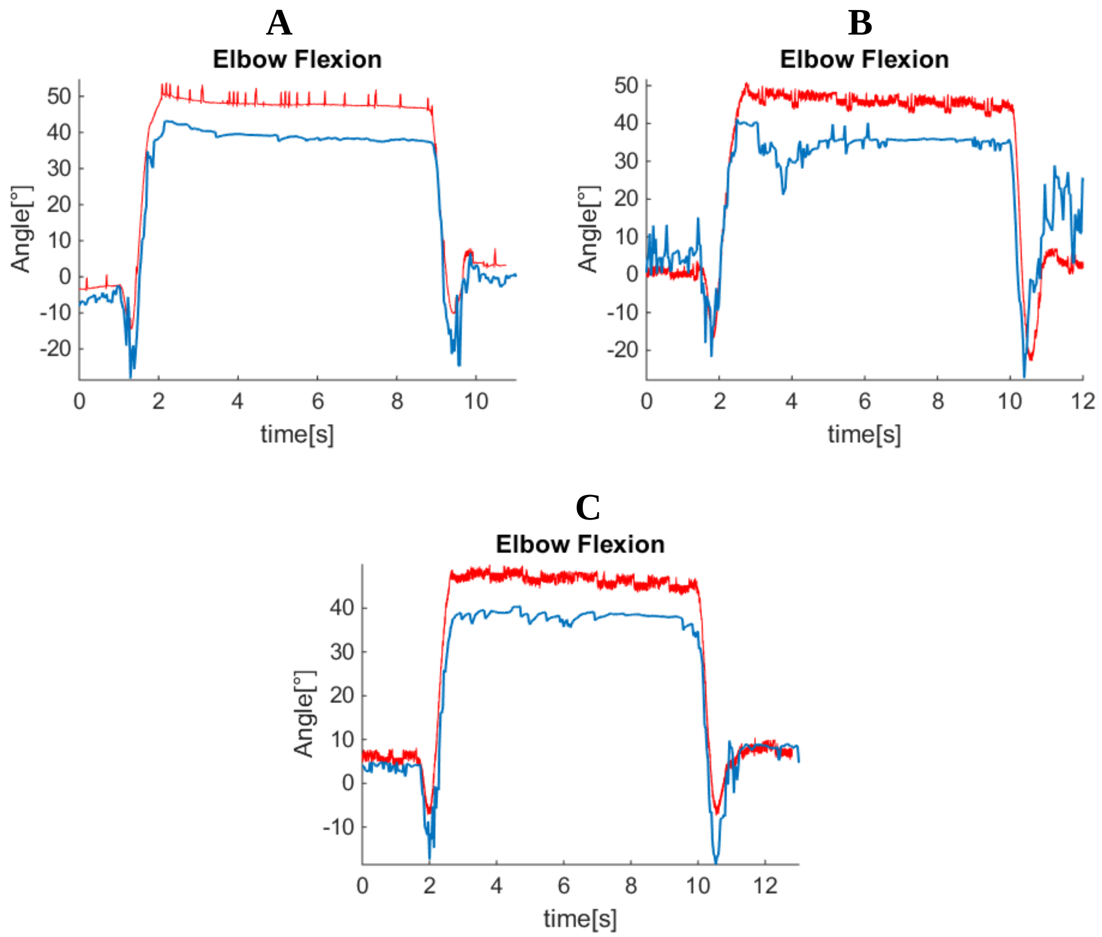



Figure 56: Elbow flexion from "full range of motions", arms upward, Kinect™ in front with desk (A), desk turned 30°-45° (B), KinectTM in front no desk (C)

In all three measurements (Figure 56) it can be seen, that the recorded data of the Kinect™ sensor are similar to the data of the goniometer, although the maximum amplitudes are smaller. Additionally, it is noticeable, that the measurement with the turned table (Figure 56, B) consists of jumpy and irregular values, compared to the other two measurements (Figure 56, A, C). It also consists of a negative peak during the stretching of the arms. Further, the second measurement (Figure 56, B) has a different starting value (~5°), while the other two measurements (Figure 56, A, C) start nearly at the same point (~-5°), however, the maximum amplitude remains the same for all three measurements (~40°).

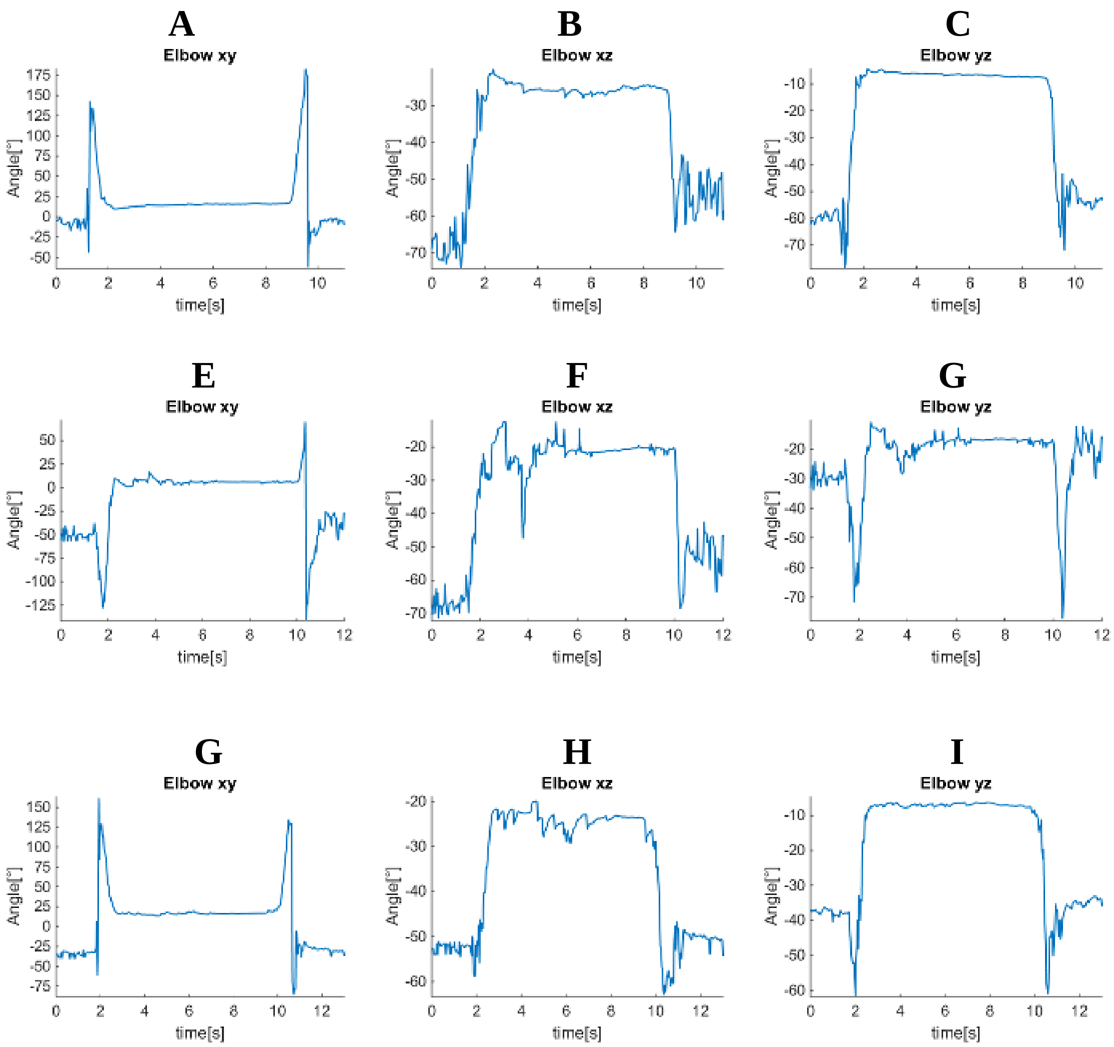



Figure 57: Projection of the elbow flexion for all three planes from “full range of motions”, arms upward, Kinect$^{TM}$ in front with desk (A, B, C), desk turned 30°-45° (D, E, F), Kinect$^{TM}$ in front no desk (G, H, I)

Looking at the graph of the single planes of the different measurements, it is noticeable, that the measurement with the turned table (Figure 57, D, E, F) looks different from the other two measurements (Figure 57 A, B, C) (Figure 57, G, H, I). Although, there are also some differences between the first & third measurement, such as the starting value or the jumpiness (cf. Figure 57, B, H) of the data, but these are only marginal. Comparing the xy- planes, large negative peaks occur within the data of the second measurement (Figure 57, D), while these peaks are positive within the other two (Figure 57, A, G). Additionally, it can be seen in the xz-plane (Figure 57, E), that a negative peak arises during the stretching of the arm, but not in the other results (Figure 57, B, H). Finally, it should be mentioned, that two negative peaks occur within the yz-plane in all three measurements, the peaks are most evident in the second measurement (Figure 57, F), followed by the third one (Figure 57, I) and slightly in the first (Figure 57, C).

# Discussion

This chapter discusses the results obtained from the Microsoft Kinect™ sensor of the performed measurements (gait analysis, hand motion & grasping) and further, the usage of the Kinect™ sensor for assessment of motor tasks.

## Microsoft Kinect™ Sensor for Gait Analysis

The aim of the gait analysis was to evaluate the quality, reliability & accuracy of the data obtained from the Kinect™ sensor. It was found, that the positioning of the Kinect™ is an important factor for the reliability & accuracy of the recorded data.

Due to the lateral positioning the legs of the proband were crossing each other during the measurement, causing misinterpretations of the joints of the legs. The Microsoft Kinect™ sensor generates a depth image of the surroundings to differentiate between objects (Zhang, 2012). Because of the lateral positioning the front (left) leg concealed the rear (right) leg, thus no tracking of the rear (right) leg could be done (Figure 58).

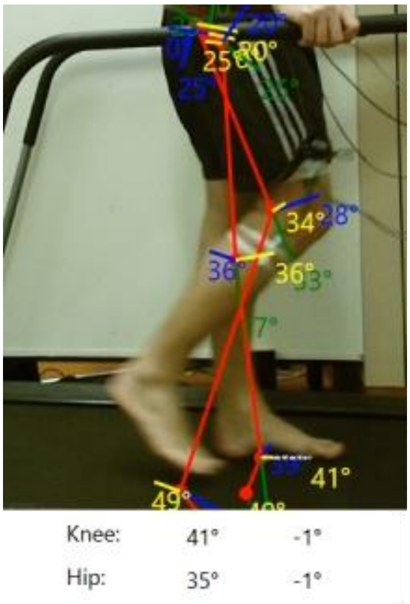



Figure 58: Failure of the Kinect™ sensor, due to the crossing of the legs (mirrored).

As result, the joints of both legs were mixed up, leading to the detection of incorrect positions and to the calculation of incorrect angles (knee, hip) (Figure 58).

Another problem was the loss of information due to the low framerate of the Kinect™ sensor. While the goniometer provided 1000 data points per second, the maximum of the Kinect™ was 30 frames per second. Consequently, the maxima of the peaks could sometimes not be detected and the form of the peaks was inhomogeneous (Figure 59).

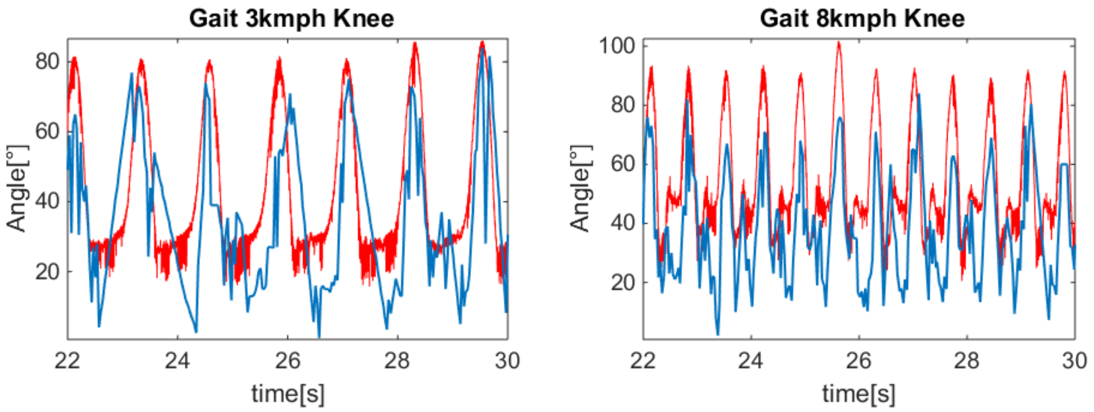



Figure 59: Difference between the data of the Kinect™ with 30fps (blue) & the data of the goniometer with 1000pps (red) at different walking speed

Thus, no declaration about the step size could be made, only a distinction of different movement speeds was possible (higher number of peaks; width of the peaks is smaller) (Figure 59).

Based on these statements it can be said, that the positioning of the Kinect™ sensor in front of the proband should lead to better results (no crossing of the legs), but the low framerate makes the use of the Kinect™ for gait analysis insufficient at the moment.

# Microsoft Kinect™ Sensor for Hand Motion & Grasping

During the hand motion and grasping tasks, the proband had to fulfill different exercises with the arms, such as striking a pose or moving an object from one position to another one. Thereby, the accuracy, reliability and limitations of the Microsoft Kinect™ sensor should be determined.

In this case it was discovered, as well as described in the previous measurement (gait analysis), that the position of the Kinect™ sensor is very important. The proband was sitting in front and with an angle of 30°-45°, while performing the tasks, which improved or worsened the recorded data (Figure 60). This can be explained by the fact, that parts of the right arm were hidden, due to the performed motion and the positioning.

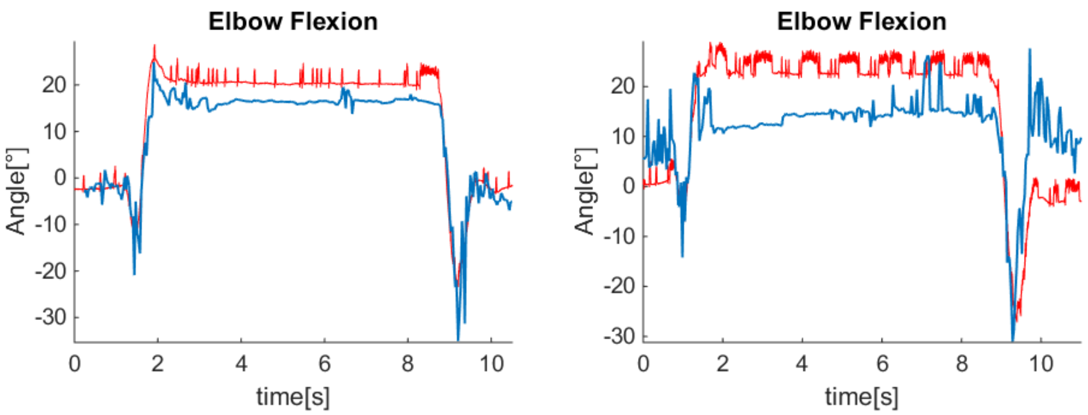



Figure 60: Recorded Data of the task "arm crossing above the head" are showing the change, due to the different positioning of the Microsoft Kinect™ sensor

The flexion of the elbow from the task "arm crossing above the head" is shown as an example (Figure 60). Due to the change of the position of the table (turned 30°-45°), the data of the Microsoft Kinect™ sensor (blue graph) got deteriorated. As a result, the number of peaks increased, the measured values for the angle decreased and the shape of the data got more distorted.

Further, it was found, that the usage of objects influences the depth perception of the Microsoft Kinect™ sensor. The depth image is determined through different flight times of the infrared beams, which are radiated from the infrared camera and then detected by the infrared sensor (Sarbolandi, Lefloch, & Kolb, 2015). Thus, it may happen, that two closed objects are interpreted as one object.

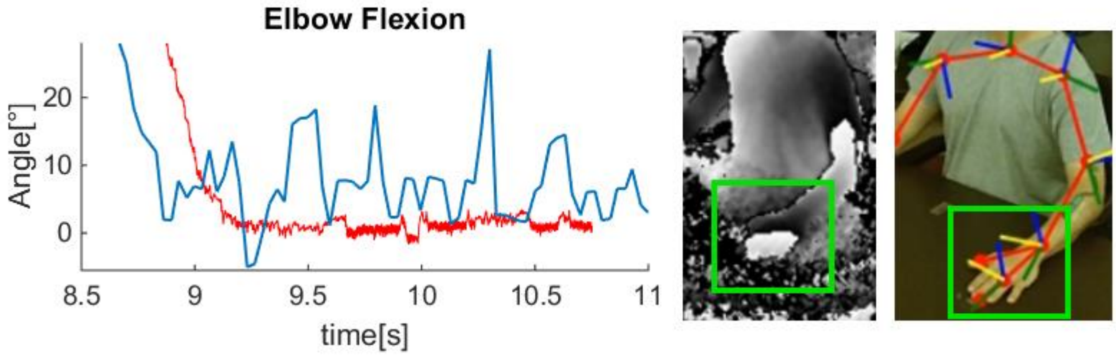



Figure 61: Recorded data of the task “arm forward” are showing the jumpiness of the calculated angles, due to the indistinguishable between the table and the hands

During the initial hand position, in which the hands were placed on the table (Figure 61), the Kinect™ sensor was not able to distinguish between the table and the hands, leading to wrong calculated angles (Figure 61), because of the jumpiness of the hand positions, visible within the blue graph.

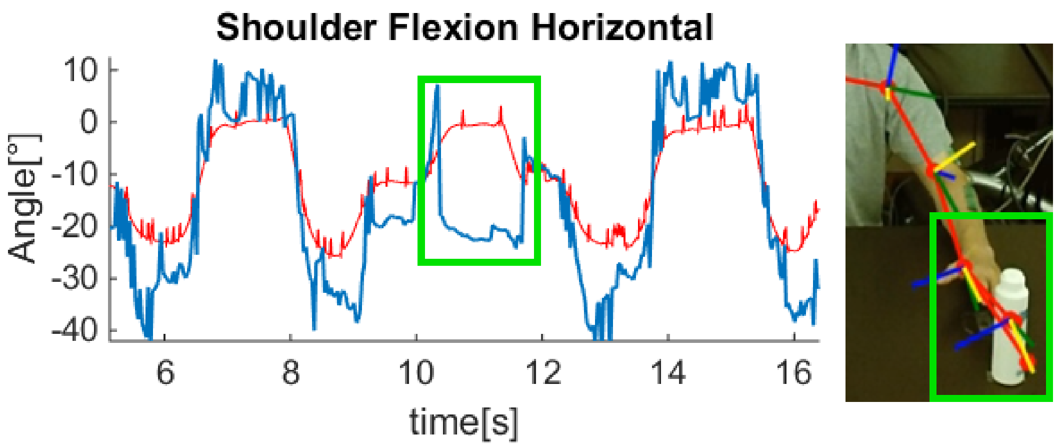



Figure 62: Recorded data of the task “grasping an object - bottle” are showing the miscalculation of the Microsoft Kinect™ sensor, due to the used object

Also, in the category “grasping an object” the object was considered as a disturbance, leading to incorrect tracking of the positions of the arm. On the one hand, the object was detected as part of the arm, due to the small depth difference, while the hand was approaching or pulled back (Figure 62). And on the other hand, the object made the tracking more difficult, because it concealed the hand temporarily.

Another limitation of the Kinect™ sensor is the detection of the arm during crossovers, as it was carried out within the category “arm crossing”. The obtained results show greater errors of the angles compared to the measurements without crossing (Figure 63). This is caused by the crossing itself, leading to a concealment of the rear arm, that makes the differentiation between the points of the arms for the Microsoft Kinect™ sensor more difficult.

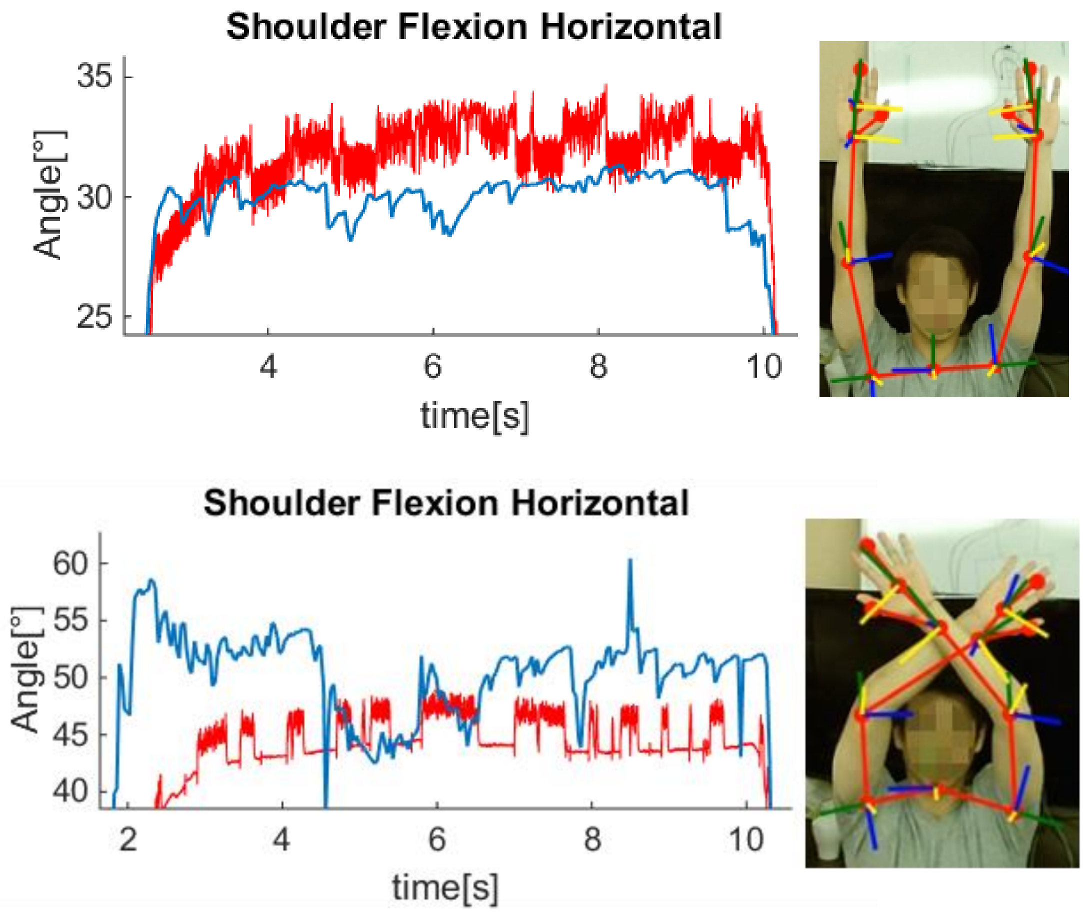



Figure 63: Differences in the error of recorded data, due to the crossing of the arms

However, the data collected from the “arm crossing above the head” show better results, than those collected from “arm crossing in front of the body”, due to the additional body points (mid of the spine, shoulder), which were hidden during the crossing (Figure 64).

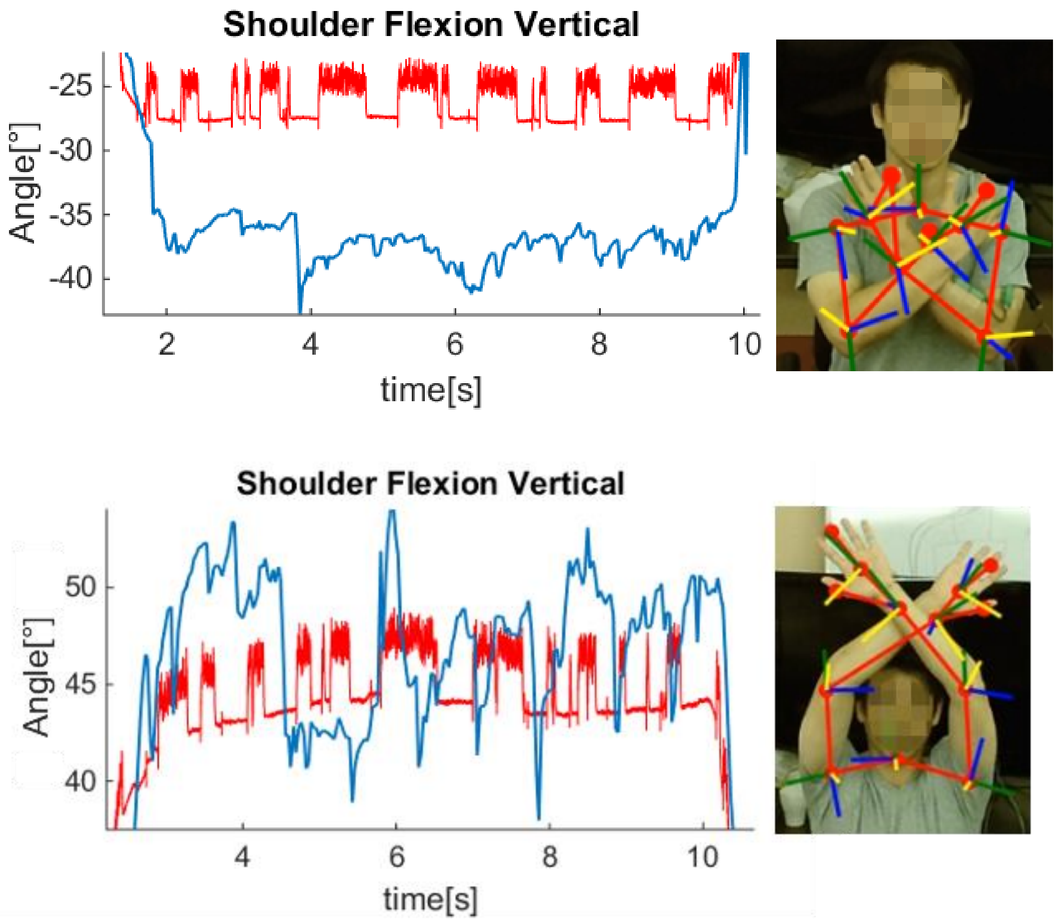



Figure 64: Depending on the type of crossing, the error of the recorded data can be smaller or higher

Furthermore, it could be determined, that the data of the Kinect™ sensor are jumpier during greater movements, than during smaller movements. The reason is, that the Kinect™ provides 30 frames per second (FPS), therefore, the resolution decreases during large and fast movements, but doesn’t change during slow and small movements.

Based on these statements, it can be said, that the use of the Microsoft Kinect™ sensor for hand motion and grasping tasks, is more suitable for small and slow movements, rather than for large and fast ones. Moreover, the positioning of the Kinect™ sensor is important, since this can greatly affect the results of the measurement. Additionally, exercises containing crossovers between two parts of the body should be avoided. The use of objects during a measurement should be prevented or chosen in a way, that they are not influencing the results of the measurement (small or transparent objects).

Since the results of the shoulder contains so much variety within each measurement, the reliability and the accuracy, about the data themselves are questionable. In order to prove the righteousness, the method to calculate the shoulder angle was used to estimate the projections of the elbow. Subsequently these projections were compared with each other (Figure 65).

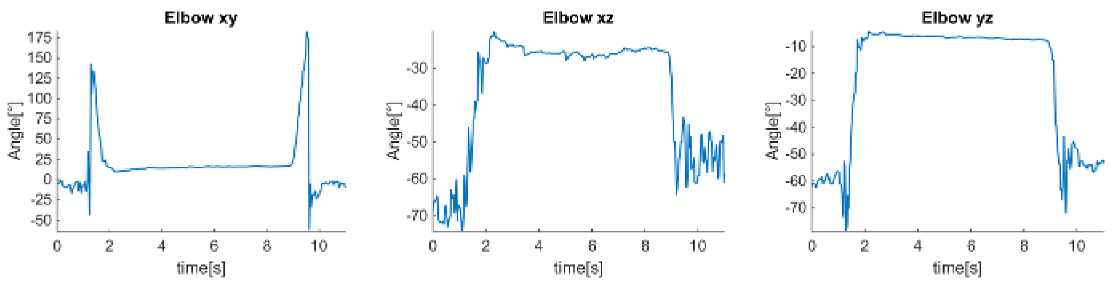



Figure 65: Projection of the elbow flexion for all three planes from “full range of motions”, arms upward, Kinect™ in front with desk

First, it was assumed, if the elbow is moving in space with a constant angle, then the angle for all three projections must change, due to a relationship between these three planes.

When evaluating the results, it was found out, that a change in one plane correspond to a change from another plane and vice versa (Figure 65). The fact, that this relationship is still present and not affected by the calculation method and the estimated values are plausible, the calculation was considered as correct.

# Conclusion

This thesis is concerned with the issue whether the new generation of the Microsoft Kinect™ sensor is capable to record three dimensional motion for assessment of motor tasks in medical applications.

First of all, the Kinect™ sensor was checked, whether it contains the necessary functions and parameters to meet the requirements. Subsequently, a program was implemented, which processes and stores the data of the Kinect™ sensor for evaluation.

To verify the usage, different measurements (gait analysis, arm motion & grasping) were performed to ensure the accuracy, the reliability and the limitations of the Microsoft Kinect™ sensor.

From the obtained results, it could be ascertained, that different motions are detectable, but no exact values can derive therefrom. Additionally, the accuracy highly depends on external parameters, such as positioning, the use of objects during the measurement, objects within the room and on the performed motion itself.

Thus, it can be said, that the new generation of the Kinect™ sensor is a very powerful tool with a strong technology behind. However, it is very limited to the fact, that it was only designed as a gaming device. For usage in analyzation of motor tasks, it is not suitable in the current state.

# References

Aggarwal, J. K., & Cai, Q. (1997). Human motion analysis: A review. *IEEE Nonrigid and Articulated Motion Workshop*, *73*(3), 428–440. http://doi.org/10.1002/ps.2158

Alabbasi, H. A., Moldoveanu, P. F., & Moldoveanu, P. A. (2015). Real Time Facial Emotion Recognition using Kinect V2 Sensor, *17*(3), 61–68. http://doi.org/10.9790/0661-17326168

Amon, C., & Fuhrmann, F. (2014). Evaluation of the Spatial Resolution Accuracy of the Face Tracking System for Kinect for Windows V1 and V2. *6th Congress of Alps-Adria Acoustics Assosiation*, (October), 9–12.

Ashley, J. (2014). Quick Reference: Kinect 1 vs Kinect 2. Retrieved July 24, 2016, from http://www.imaginativeuniversal.com/blog/post/2014/03/05/Quick-Reference-Kinect-1-vs-Kinect-2.aspx

Begg, R., & Palaniswami, M. (2006). *Computational intelligence for movement sciences neural networks and other emerging techniques*. *Computational intelligence and its applications series*. Hershey, PA, USA: IGI Global. http://doi.org/10.4018/978-1-59140-836-9

Brown, B. H., Smallwood, R. H., Barber, D. C., Lawford, P. V, & Hose, D. R. (1999). Medical physicis and biomedical engineering. *Cellular Engineering*, 736.

C. Sankaranarayanan, A., Chellappa, R., & Baraniuk, R. G. (2011). Distributed Video Sensor Networks. *Distributed Video Sensor Networks*, 85–101. http://doi.org/10.1007/978-0-85729-127-1

Ceseracciu, E., Sawacha, Z., & Cobelli, C. (2014). Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: Proof of concept. *PLoS ONE*, *9*(3), 1–7. http://doi.org/10.1371/journal.pone.0087640

Gavrila, D. M. (1999). The Visual Analysis of Human Movement: A Survey. *Science Direct*, *73*(1), 82–98. http://doi.org/10.1006/cviu.1998.0716

Huang, Y., & Huang, T. S. (2002). Model-based human body tracking. *Proceedings. 16th International Conference on Pattern Recognition*, *1*, 552–555. http://doi.org/10.1109/ICPR.2002.1044791

Microsoft. (2016a). Coordinate mapping. Retrieved July 24, 2016, from https://msdn.microsoft.com/en-us/library/dn785530.aspx

Microsoft. (2016b). JointType Enumeration. Retrieved July 24, 2016, from https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx

Mousavi Hondori, H., & Khademi, M. (2014). A Review on Technical and Clinical Impact of Microsoft Kinect on Physical Therapy and Rehabilitation. *Journal of Medical Engineering*, *2014*, 1–16. http://doi.org/10.1155/2014/846514

Nos, T. (1998). Goniometer and torsiometer operating manual. *Biometrics Ltd.*, 1–24.

Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, *108*(1-2), 4–18. http://doi.org/10.1016/j.cviu.2006.10.016

Sarbolandi, H., Lefloch, D., & Kolb, A. (2015). Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect. *CoRR*, *abs/1505.05459*. Retrieved from http://arxiv.org/abs/1505.05459

Soares Beleboni, M. G. (2014). A brief overview of Microsoft Kinect and its applications. *Interactive Multimedia Conference 2014*, 6. Retrieved from http://mms.ecs.soton.ac.uk/2014/papers/2.pdf

Stathopoulou, I.-O., & Tsihrintzis, G. A. (Eds.). (2010). *Visual Affect Recognition*. Amsterdam, The Netherlands, The Netherlands: IOS Press. Retrieved from http://dl.acm.org/citation.cfm?id=1875466

Su, C.-J. (2013). Personal Rehabilitation Exercise Assistant with Kinect and Dynamic Time Warping. *International Journal of Information and Education Technology*, *3*(4), 448. http://doi.org/http://dx.doi.org/10.7763/IJIET.2013.V3.316

Wang, L., Hu, W., & Tan, T. (2003). Recent developments in human motion analysis. *Pattern Recognition*, *36*(3), 585–601. http://doi.org/10.1016/S0031-3203(02)00100-0

Weber, I., Koch, J., Meskemper, J., Friedl, K., Heinrich, K., & Hartmann, U. (2012). Is the MS Kinect suitable for motion analysis? *Biomedical Engineering. Biomedizinische Technik*, *57*(SI-1), 664. http://doi.org/10.1515/bmt-2012-4452

Wei, X., Zhang, P., & Chai, J. (2012). Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph.*, *31*(6), 188:1–188:12. http://doi.org/10.1145/2366145.2366207

Whittle, M. W., & Whittle, M. W. (2007). Preface to fourth edition. In *Gait Analysis*. http://doi.org/10.1016/B978-075068883-3.50004-0

Zennaro, S., Munaro, M., Milani, S., Zanuttigh, P., Bernardi, A., Ghidoni, S., & Menegatti, E. (2013). PERFORMANCE EVALUATION OF THE 1ST AND 2ND GENERATION KINECT FOR MULTIMEDIA APPLICATIONS S . Zennaro , M . Munaro , S . Milani , P . Zanuttigh , A . Bernardi , S . Ghidoni , E . Menegatti.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE Multimedia*, *19*(2), 4–10. http://doi.org/10.1109/MMUL.2012.24