

Compute rotation around the hinge between the frame and the steer

Get axis of rotation

We computed the axis of rotation in the callibration trial using a least-squares approach [Ehrig2007]. We can compute the rotation of the steer in the coordinate system of the frame R_{frame}^{steer} from the orientation of the frame in the world R_{world}^{frame} and of the steer in the world R_{world}^{steer} . These orientation of the world are based on the sensor fusion in Xsens. (Note: we attached one sensor the the frame of the bike and one sensor to the steer of the bike. We assume that all rotation occurs around a hinge representing the stem of the bike)

$$R_{frame}^{steer} = R_{world}^{frame^T} * R_{world}^{steer} \quad (1)$$

Hence we can compute the function axis in the steer (η_{steer}) and in the frame (η_{frame}) by solving this equation in a least-squares sense (we have n-frames for R_{steer}^{frame}). We solved the least-squares problem using a singular value decomposition in matlab (using function svd).

$$\eta_{frame} = R_{steer}^{frame} \eta_{steer} \quad (2)$$

In the next step we want to compute the rotation around this axis. Therefore, we adapted the coordinate system (R) of the frame such that the x-axis aligns with η_{frame} .

$$z = n_{frame} \times [0 \quad 1 \quad 0] \quad (3)$$

$$y = z \times n_{frame} \quad (4)$$

$$R = [n_{frame} \quad y \quad z] \quad (5)$$

As a result we can compute the rotation around the axis of rotation in a rotation matrix (R_{axis}) and convert it to euler angles ($\theta_{stuurRot}$)

$$R_{axis} = R * R_{frame}^{steer} \quad (6)$$

$$\theta_{stuurRot} = rotm2eul(R_{axis}) \quad (7)$$

Result

This works well in most cases. In figure 1 you can see that the error on equations 2 is low during the callibration, there is only movement along the stem of the bike during the callibration trial. Futhermore, we steering angle looks fine during the whole movement.

In some trials, mainly with the normal bike, we see also rotation around the y and z axis, which we don't expect. On the one hand, this might be related to different sensor placement and the stem of the bike. On the other hand, the rotation is too large to be explained by a loose stem of the bike...

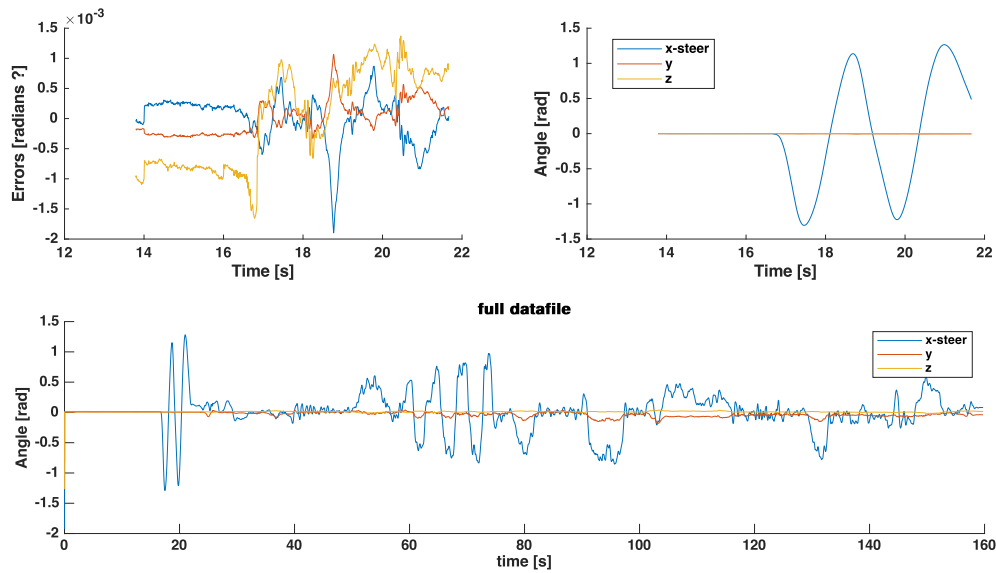


Figure 1: ...

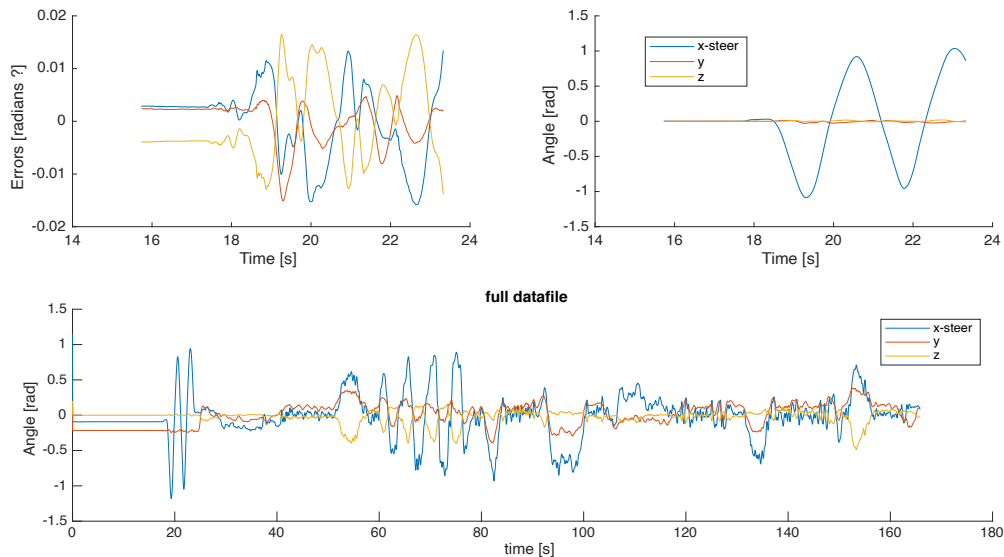


Figure 2

Batch processing in matlab

You have to run the script *GetRotationAxisSteer_Subjects.m* to:

- Compute the axis of rotation in each subjects with the normal bike and the classic bike
- Compute the steering angle in all subjects and trials.

These scripts mainly rely on the functions

- *getCallibrationPhase.m* automatically detect the callibration phase
- *GetHingeAxis.m* to compute the rotation axis
- *GetAngleSteer.m* to compute the steering angle
- *InterpolateRotMatrices.m* to interpolate the orientations of the sensors when frames are missing (not very common). We first convert to rotation matrices to euler angles, interpolate the euler angles and convert to rotation matrices again.

References

Ehrig2007: A survey of formal methods for determining functional joint axes