The development and evaluation of a sensor-fusion and adaptive algorithm for detecting real-time upper-trunk kinematics, phases and timing of the sit-to-stand movements in stroke survivors

Presenter: Siu Ho (4th year, Doctor of Engineering)
Other authors: Dr Andy Kerr, Dr Avril Thomson
Background

• Stroke
  • The largest cause of complex disability in adults (Adamson, Beswick, & Ebrahim, 2004)

• Sit-to-stand (STS)
  • Critical to activities of daily living

• Physical Rehabilitation
  • Regain functional movement

• Assign treatment plan

• Assess performance

(Janssen et al, 2010)
Clinical assessments

- Five times STS test (NHS)
- 30 Seconds Chair Stand Test
- Timed Up and Go test

- Visual Observations

- Accurate to quantify
  - Manually timed
  - Repetitions

- Inexact to characterise
  - Biomechanical performance
  - Weight symmetry loading, velocity, angles
Adopting Technology

Analysis Performance

- Vicon motion capture
  - Markers + Large space for infrared red cameras
- Kinect
- Fixed force plate

Problems

- Restricted to Lab
- Time consuming
- Set-up
- “Unobstructable”
Hypothesis

Wearable Technology

- Inertial Measurement Unit (IMU)
  - Low-cost
  - Mini-natured
  - Plug-and-Play
  - Low power and high performance

Developed Algorithm

- Healthy Individuals  (Cerrito, Bichsel, Radlinger, & Schmid, 2014)
- Elderlies  (Guimaraes, Ribeiro, & Rosado, 2013)
- Other disorders  (Zijlstra, Mancini, Lindemann, Chiari, & Zijlstra, 2012) (Van Lummel et al., 2012)
- No stroke
Sit-to-stand Event Detection

1. Initiation
2. Seat-off
3. Flexion
4. Stand

Phase 0
- Start
- Wait
- Feet on plate & Trunk rolled forward by 5°

Phase 1
- Feet off plate or Trunk angle drops below 5°
- User returning to chair

Phase 2
- Trunk angle reduces from 30°
- Trunk rolled 30° & shifting weight to feet
- User returning to chair

Phase 3
- Trunk angle returns to 0° & full body weight on plate

Phase 4
- End
- User returning to chair
Orientation Estimation

Accelrometer

- Measuring g-forces
- Trigonometry to find inclinations
- External acceleration 😞
- No drifts 😊

Gyroscope

- Not affected by external acceleration 😊
- Integration drift ☹️
Kalman Filtering

Sensor Fusion Algorithm
- Mixing data from accelerometer and gyroscope
- Observe measurements (noise/inaccuracies)

Time Update
1. Project state ahead
   \[ \theta_{\text{est } k} = A \theta_{\text{est } k-1} + B u_k + w_{k-1} \]
2. Project Error covariance (P)
   \[ P^-_k = AP_{k-1}A^T + Q \]

Measurement Update
1. Calculate Kalman Gain (K)
   \[ K_k = P^-_k H^T (HP^-_k H^T + R)^{-1} \]
2. Update estimate
   \[ \theta_{\text{est } k} = \theta_{\text{est } k} + K_k (\theta_{\text{gyro}} - H \theta_{\text{acc } k}) \]
3. Update error covariance
   \[ P_k = (1-K_k H) P^-_k \]

A = State transition matrix
B = Optional control matrix
\( u_k \) = known system inputs
\( w_{k-1} \) = process noise vector
Q = covariance matrix
H = system observation matrix
\( ^T \) = Transpose
Velocity Estimation

**Accelerometer**
- Integration drift 😞
  - Accumulation of errors
- Need to remove gravity
  - Gravity offset – Inclination
  - Sensitivity

**Balance-plate**
- Centre of pressures
- Predict velocity
  - Need acceleration estimation
Sensor Fusion Algorithm

Gyroscope
- Sample delay (by n=10)
- Tilt compensation for calibration
- Integration to obtain angular displacements

Accelerometer
- Butterworth Low-Pass Filter
- Sliding Average Filter (n=10)
- Calculate angle Inclinations

Kalman Filter: Estimate upper-trunk orientation

Balance-Plate
- Butterworth Low Pass Filter
- Sample delay (by n=10)
- Differentiation to obtain rate of change of CoP’s

Detect and predict change in upper-trunk velocities

Sensitivity compensation (Project accelerations to global axis) and correct gravity

Integration to obtain velocities

Final results: Estimate the actual velocities

Obtain upper-trunk orientation

Obtain upper-trunk velocities
Implementation and Testing

Capture STS performance via Vicon and Sensors
• Processed on Vicon Nexus and MatLab
• Design algorithms and filters + simulations

Performance Algorithms V.S. Vicon
• LabView (C, Mathscript code)
Results – Angle Estimation

**Accelerometer**
- Under and overestimated
- Linear acceleration
- Gravity

**Gyroscope**
- Overestimated
- Inaccuracy in raw signal
- Integration drifting

**Sensor-Fusion**
- Close estimate
- Delay (filtering)
Results – Velocity Estimation

Slow motion
- Closely matched
- Smoothed by filters

Quick motion
- Small systematic bias
- Lower mean and peak vertical velocity
- Illegible with slower STS
Discussion

Further Improvement

• Inconsistent sampling rate
• Better IMU
• More stroke survivors involve (e.g. those who can’t stand-up)
• Diagnostic platform
  • Feedback on performance

Conclusion

• New approach in tracking STS movement
  • Sensor-Fusion
  • Finite State Machine
• Validated, Vicon and stroke survivors
• Estimate, Track and Analyse
Thank you so much for your attention!

Any Questions?