INS/EKF-Based Stride Length, Height and Direction Intent Detection for Walking Assistance Robots

Dario Brescianini, Jun-Young Jung, In-Hun Jang, Hyun Sub Park and Robert Riener

Abstract—We propose an algorithm used to obtain the information on stride length, height difference, and direction based on user’s intent during walking. For exoskeleton robots used to assist paraplegic patients’ walking, this information is used to generate gait patterns by themselves in on-line. To obtain this information, we attach an inertial measurement unit (IMU) on crutches and apply an extended kalman filter-based error correction method to reduce the phenomena of drift due to bias of the IMU. The proposed method is verified in real walking scenarios including walking, climbing up-stairs, and changing direction of walking with normal.

I. INTRODUCTION

Exoskeleton robots are increasingly used as mobility aids since they offer many advantages over conventional devices. By supporting patients while walking in a standing posture with their own legs and taking part in ordinary daily activities, exoskeleton robots can enhance a patient’s quality of life and self-esteem.

Several wearable exoskeleton robots for paraplegic patients have been developed. One of the key elements of these walking support systems is their user interface. The interface should be able to determine the users walking intention: starting and stopping of walking, stride length, height difference, direction, and operation mode (i.e., walking straight or climbing stairs). Furthermore, the interface should be very intuitive and natural. If the exoskeleton robot could synchronize patient’s intentions with robot’s movements, then the patient could feel comfortably in control of the robot. Conventionally, exoskeletons estimate user intentions by measuring biological signals such as myoelectricity [1]. However, this is not possible for spinal-cord injured (SCI) patients because the biological signals of SCI patients do not reach the lower body. Therefore, new methods must be found. HAL [2, 3] measures the floor reaction force, which resembles the center of gravity, to detect weight shifting between the legs and uses this as an input to trigger a stride. However, it is not possible with this method to estimate the users intended stride length, height difference, or direction. ReWalk [4, 5] uses a gesture-based human-machine interface to estimate the patient’s intention. ReWalk derives the user’s intention from the torso tilt that takes place before each step. However, ReWalk also does not have a function that can estimate the user’s intent regarding stride length, height difference, and direction in walking. One common property of exoskeleton robots is that crutches are needed to maintain stability in their upright posture.

In this paper, we propose an inertial navigation system (INS) and extended Kalman filter (EKF)-based estimation algorithm to obtain the information regarding stride length, height difference, and direction from the user’s intent in a walking situation. In order to develop the algorithm, we assumed the following: (1) The crutches are moved ahead before each step to ensure stability. (2) The distance the crutches are moved ahead and the stride length are related (e.g. if the user wants to move their legs in a long stride, then they must move their crutches farther). (3) The exoskeleton robot already has two functions: walking start detection based on a walking stability criterion, and a gait pattern generator. Therefore, the functions of these are not dealt in this paper.

II. MATERIALS AND METHODS

The user intent detection system is shown in Fig. 1. The system can be split into two parts: a crutch trajectory estimator for detecting the gait parameters, and a weight shift detector (which is not discussed in this paper) that is responsible for triggering a step with the estimated gait parameters.

A. Sensor System

In order to measure the crutch motion, an inertial measurement unit (IMU) is mounted on the crutch shaft. The IMU is placed such that the x-axis is collinear to the shaft. However, most small inertial sensing devices suffer from large drifts and it is therefore not sufficient for use as the only input for motion detection. For this reason, a force sensing resistor (FSR) is placed between the crutch tip and the shaft to detect when the crutch touches the ground, and to correct the drift.
B. Gait Parameter Estimation

An extended Kalman filter (EKF)-based inertial navigation system (INS) algorithm was implemented following the approaches described in [6, 7] to compute the crutch trajectory. In a classic INS mechanization, velocity and position errors will diverge with time due to drift in gyroscopic readings. The idea of an EKF-based INS algorithm is to correct the velocity and position estimate and reduce errors with an EKF using ground contact information. Every time the crutch touches the ground, the crutch tip has zero velocity and the estimated velocity can thus be corrected.

1) INS mechanization: The body frame coincides with the axis of the IMU. The navigation frame is initialized before each step with the x-axis collinear to the sagittal plane, the y-axis in the lateral plane, and the z-axis upward (Fig. 2). The reason for this initialization is based on the assumption that the crutch tip’s movement length along the sagittal plane, the y-axis collinear to the sagittal plane, and the z-axis in the lateral plane, and the x-axis are equal. Therefore, the body frame and the axis of the IMU coincide after each step. The INS mechanization is divided into two stages. The first stage is conducted before the EKF correction. After the EKF correction, the second stage is executed to get the corrected position of the crutch tip.

Let $C_{b}^{n}$, $v^{n}$, $\omega^{b}$, $a^{b}$, and $g$ be defined as

- $C_{b}^{n} \in R^{3 \times 3}$: direct-cosine matrix that transforms from the body frame $b$ to the navigation frame $n$.
- $v^{n} \in R^{3 \times 1}$: position in the navigation frame.
- $\omega^{b} \in R^{3 \times 1}$: gyroscope output.
- $a^{b} \in R^{3 \times 1}$: accelerometer output.
- $g \in R^{3 \times 1}$: gravitational acceleration in the navigation frame.

In the first stage, the raw inertial sensor data is bias compensated based on Kalman bias estimates (23) for the gyroscopes $\delta\omega^{b}$ and the accelerometers $\delta a^{b}$ respectively:

$$\omega_{k}^{b} = \omega_{k}^{b} + \delta\omega_{k-1}^{b}$$
$$a_{k}^{b} = a_{k}^{b} + \delta a_{k-1}^{b}$$  (1)

The bias-corrected angular rates are then integrated to estimate the orientation, and the accelerations are double-integrated yielding a position estimate:

$$\hat{C}_{b,k}^{n} = C_{b,k-1}^{n} \frac{2I_{3 \times 3} + \delta\Omega_{k} \cdot \Delta t}{2I_{3 \times 3} - \delta\Omega_{k} \cdot \Delta t}$$
$$\hat{v}_{k}^{n} = \hat{v}_{k-1}^{n} + \left[ \hat{C}_{b,k}^{n} \cdot a_{k}^{b} - g \right] \cdot \Delta t$$
$$\hat{r}_{k}^{n} = \hat{r}_{k-1}^{n} + \hat{v}_{k}^{n} \cdot \Delta t$$  (2)

where

$$\delta\Omega_{k} = \begin{bmatrix} 0 & -\omega_{k}^{h}(3) & \omega_{k}^{h}(2) \\ \omega_{k}^{h}(3) & 0 & -\omega_{k}^{h}(1) \\ -\omega_{k}^{h}(2) & \omega_{k}^{h}(1) & 0 \end{bmatrix}$$  (3)

and

$$\delta\varphi_{k} = \hat{C}_{b,k}^{n} \cdot \delta\varphi_{k} + \hat{C}_{b,k}^{n} \cdot \delta\varphi_{k} + \hat{C}_{b,k}^{n} \cdot \delta\varphi_{k}$$  (4)

where $\delta\omega_{k}^{b}(3)$ means the third element of $\delta\omega_{k}^{b}$.

In Eq. (3), it is assumed that the rotation rate is small, and an approximation for the exponential function is used. Finally, in this second stage, the previously computed attitude $C_{b}^{n}$, velocity $\hat{v}^{n}$ and position $\hat{r}^{n}$ are corrected with the EKF error estimations (23):

$$C_{b,k}^{n} = \frac{2I_{3 \times 3} + \delta\Theta_{k}}{2I_{3 \times 3} - \delta\Theta_{k}} \hat{C}_{b,k}^{n}$$
$$\hat{v}_{k}^{n} = \hat{v}_{k-1}^{n} + \delta v_{k}$$
$$\hat{r}_{k}^{n} = \hat{r}_{k-1}^{n} + \delta r_{k}$$  (5)

where $\delta\Theta_{k}$ is defined as in Eq. (6) with $\delta\varphi_{k}$ instead of $\omega_{k}^{b}$.

2) The extended Kalman filter: The extended Kalman filter is used to estimate attitude, position, and velocity, and $\delta\omega_{k}^{b}$ and $\delta a_{k}^{b}$ are the gyro and accelerometer biases. The original state transition model is
a non linear function of the states, but it can be linearized around a state estimate [6]. The linearized state transition model of the error dynamics is given in Eq.(11):

\[ x_k = F_k x_{k-1} + w_k \]  

(11)

where \( x_k \) and \( x_{k-1} \) are state vectors and \( w_k \) is the process noise with covariance matrix \( Q_k = E(w_k w_k^T) \). The actual prediction processes are given as

\[ \delta \hat{x}_k = F_k \delta x_{k-1} \]  
\[ \tilde{P}_k = F_k^{-1} P_{k-1} F_k^{-1} + Q_{k-1} \]

(12)

(13)

where \( P_k \) is the error covariance matrix. In [7], the state transition matrix of the error dynamics at time \( k \) is given as

\[ F_k = \begin{bmatrix}
I & \Delta t \cdot \tilde{C}_{b,k}^n & 0 & 0 & 0 \\
0 & I & 0 & 0 & 0 \\
-\Delta t \cdot S(a_{b,k}^n) & 0 & I & \Delta t \cdot \tilde{C}_{b,k}^n & 0 \\
0 & 0 & 0 & 0 & I \\
\end{bmatrix} \]  

(14)

The term \( S(a_{b,k}^n) \) is a skew matrix that causes attitude errors to affect velocity errors, and allows the EKF to act as an inclinometer to provide corrections to the orientation estimate:

\[ S(a_{b,k}^n) = \begin{bmatrix}
0 & -a_{b,k}^n(3) & a_{b,k}^n(2) \\
a_{b,k}^n(3) & 0 & -a_{b,k}^n(1) \\
a_{b,k}^n(2) & a_{b,k}^n(1) & 0 \\
\end{bmatrix} \]  

(15)

where \( a_{b,k}^n \) is the bias-corrected acceleration expressed in the navigation frame.

The measurement model for the error-state filter is:

\[ z_k = H \delta x_k + n_k \]  

(16)

where \( H \) is the measurement matrix and \( n_k \) is measurement noise with covariance matrix \( R_k = E(n_k n_k^T) \).

If the FSR in the crutch detects ground contact, the EKF can be fed with pseudo measurements of the velocity error. If the crutch is assumed to be a rigid body, the velocity can be computed as

\[ \dot{v}_k^n = \vec{C}_{b,k}^n \cdot \dot{v}_k^b = \vec{C}_{b,k}^n \cdot (\omega_k^b \times \dot{v}_k^b) \]  

(17)

where \( \dot{v}_k^b \) is the vector pointing from the crutch tip to the IMU. The measured velocity error is therefore given as

\[ \Delta v_k^n = \dot{v}_k^n - \dot{v}_k^n \]  

(18)

The yaw orientation is not observable from the velocity measurements in Eq.(18). To reduce drift in the heading, we further assumed that the rate of rotation around the shaft, which corresponds to the axis \( e_3^b \), is very small while touching the ground. Thus, it is impossible to rotate much this shaft when the crutch tips are touching the ground with the user’s weight except in a falling situation. This assumptions yields the measured error in angular rate to be

\[ \Delta \omega_k^i = \omega_k^i(1); \]  

(19)

Combining Eqs.(18) and (19) to the error measurement vector yields

\[ m_k = \begin{bmatrix}
\Delta \omega_k^i \\
\Delta \omega_k^i \\
\end{bmatrix} \]  

(20)

Then, the measurement matrix must be

\[ H = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
\end{bmatrix} \]  

(21)

If a measurement is available; i.e., if the crutch is on the ground, the estimates can be updated with

\[ \delta \hat{x}_k = \delta \hat{x}_k + K_k \cdot [m_k - H \delta \hat{x}_k] \]  

(22)

\[ P_k = (I - K_k H) \tilde{P}_k (I - K_k H)^T + K_k R_k K_k^T \]  

(23)

where \( K_k \) is the Kalman gain computed with the standard formula:

\[ K_k = \tilde{P}_k H^T \left( H \tilde{P}_k H^T + R_k \right)^{-1} \]  

(24)

It should be noted that all non-bias terms in the error state vector \( \delta \hat{x}_k \) are cleared back to zero after each estimation cycle because these errors are compensated in the INS estimations described by Eqs.(7-9).

3) Walking Intent Determination: In order to derive the user’s walking intent from the crutch trajectory, the position of the crutch tip

\[ r_{\text{tip}}^n = r^n - C_i^n \cdot \dot{v}_k^b \]  

(25)

is computed as long as the crutch touches the ground. The measurement noise covariance matrix \( R_k \) was set so that the EKF corrects the errors gradually over 50 ms. After the patient moved the crutches ahead and ground contact is detected, the system waited 50 ms before computing the difference in crutch position:

\[ \Delta r_{\text{tip}}^n = r_{\text{tip, final}}^n - r_{\text{tip, initial}}^n \]  

(26)

If \( |r_{\text{final}}^n(3)| \geq 2 \text{ cm}; \) i.e., the crutch level was either lowered or raised more than 2 cm, it is assumed that the patient wants to climb or descend stairs in which case larger foot clearance is needed. For this case, gait parameters are determined as

\[ \text{stride length} = \sqrt{\Delta r_{\text{tip}}^n(1)^2 + \Delta r_{\text{tip}}^n(2)^2} \]  

(27)

\[ \text{direction} \alpha = \tan^{-1} \left( \frac{\Delta r_{\text{tip}}^n(2)}{\Delta r_{\text{tip}}^n(1)} \right) \]  

(28)

\[ \text{height difference} = \Delta r_{\text{tip}}^n(3). \]  

(29)

Otherwise, the patient intends to walk on the same level with the gait parameters

\[ \text{stride length} = |\Delta r_{\text{tip}}^n| \]  

(30)

\[ \text{direction} \alpha = \tan^{-1} \left( \frac{\Delta r_{\text{tip}}^n(2)}{\Delta r_{\text{tip}}^n(1)} \right). \]  

(31)

Depending on crutch gait type; i.e., on how many times per complete walking cycle the crutches are moved forward, the actual stride length may be multiples of Eqs.(27) or (30). The intended direction is expressed with a rotation angle \( \alpha \) around the axis \( e_3^n \) relative to the sagittal plane.
C. Experimental Evaluation

An ADIS16350 IMU from Analog Devices was used to perform the experiments. The IMU was mounted on a crutch at a height of 41 cm. Inertial measurements were taken at a rate of 200 Hz and the gait parameters were then computed offline using MATLAB(2010). In order to evaluate the performance of the gait parameter estimation algorithm, two different tests were performed.

III. Results

In the first series of tests, a crutch gait was executed with predefined stride lengths and directions to assess accuracy and precision. The algorithm was then applied to a real indoor scenario to verify the assumption that it is possible to derive the intended gait parameters from the crutch trajectory. For this experiment, the computed gait parameters were added and compared to the real environment. Both experiments were performed using a three-point crutch gait for which the stride length was twice the distance covered by the crutches. Figure 3 shows the estimated crutch trajectories for three different types of crutch movement: 1) climbing a step 17.5 cm in height; 2) walking with a stride length of 60 cm; and 3) walking with a stride length of 1 m.

![Fig. 3. Computed crutch trajectories for different crutch movements.](image)

The computed gait parameters for 40 performed strides of each type are summarized in Table I.

<table>
<thead>
<tr>
<th>crutch gait type</th>
<th>stride length [cm]</th>
<th>height difference [cm]</th>
<th>direction α [°]</th>
<th>time [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>μ</td>
<td>σ</td>
<td>μ</td>
<td>σ</td>
</tr>
<tr>
<td>1</td>
<td>46.32</td>
<td>2.52</td>
<td>17.84</td>
<td>1.80</td>
</tr>
<tr>
<td>2</td>
<td>60.74</td>
<td>2.04</td>
<td>0.25</td>
<td>0.22</td>
</tr>
<tr>
<td>3</td>
<td>101</td>
<td>3.14</td>
<td>0.13</td>
<td>0.76</td>
</tr>
</tbody>
</table>

TABLE I

Computed gait parameters for different gait types.

In the second result, an experiment to verify whether the proposed algorithm is effective in a continuous step situation and generates continuous crutch trajectories was conducted with the situation of walking for three steps and climbing four stairs. The results of this experiment are shown in Fig. 4. A staircase was approached from a 1m distance. It can be seen that the subject changed his stride length just before approaching the first step in order to climb the stairs comfortably.

![Fig. 4. Computed trajectory of the crutch tip for a real scenario. Each * indicates a position where the crutch touched the ground.](image)

IV. Discussion

In the first experimental result, we note that the gait parameter estimator performed well and was clearly able to detect different walking situations. Although the standard deviations for stride length and direction seem to be very large, these figures should be compared to the stride variability of a normal person. A height difference between two subsequent steps was detected very accurately by the estimator. The proposed algorithm could therefore be used to adapt a gait pattern if a patient walks on a small slope. However, this algorithm is limited to normal crutch gait walking in which the crutches leave the ground for only a few seconds. Otherwise, our results will strongly vary due to cubic-in-time error growth (Table I).

In the second experiment, the estimated crutch trajectories in Fig. 4 clearly matched the real environment. Since we expected that the exoskeleton robot would execute its motions based on these estimates, the robot movements will also conform to the environment. Due to this, it can be concluded that it is possible to derive the gait parameters from the crutch motion. Moreover, as shown in Fig. 4, errors were added up. In the real case, if the stride length was estimated as too short, the patient would then correct this in a following stride and move the crutches ahead further. Thus errors would not propagate.

V. Conclusion

We presented an algorithm that can estimate a paraplegic patient’s intended stride length, walking direction, and height difference for use with exoskeleton rehabilitation robots. The proposed algorithm uses the crutch trajectory to compute these gait parameters. This strategy is very comfortable and intuitive for the patient since the patient does not need to perform any additional task except ensuring his stability with crutches.
References


