

Ground Reaction Force Sensor Fault Detection and Recovery Method based on Virtual Force Sensor for Walking Biped Robots

Iyad Hashlamon

Faculty of Engineering and Natural Sciences
Sabanci University
Tuzla, Istanbul, Turkey
hashlamon@sabanciuniv.edu

Kemalettin Erbatur

Faculty of Engineering and Natural Sciences
Sabanci University
Tuzla, Istanbul, Turkey
erbatur@sabanciuniv.edu

Abstract—This paper presents a novel method for ground force sensor faults detection and faulty signal reconstruction using Virtual force Sensor (VFS) for slow walking bipeds. The design structure of the VFS consists of two steps, the total ground reaction force (GRF) and its location estimation for each leg based on the center of mass (CoM) position, the leg kinematics, and the IMU readings is carried on in the first step. In the second step, the optimal estimation of the distributed reaction forces at the contact points in the feet sole of walking biped is carried on. For the optimal estimation, a constraint model is obtained for the distributed reaction forces at the contact points and the quadratic programming optimization method is used to solve for the GRF. The output of the VFS is used for fault detection and recovery. A faulty signal model is formed to detect the faults based on a threshold, and recover the signal using the VFS outputs. The sensor offset, drift, and frozen output faults are studied and tested. The proposed method detects and estimates the faults and recovers the faulty signal smoothly. The validity of the proposed estimation method was confirmed by simulations on 3D dynamics model of the humanoid robot SURALP while walking. The results are promising and prove themselves well in all of the studied fault cases.

Index Terms— Quadratic programming, ground reaction forces, inertial measurement unit (IMU), virtual force sensor, sensor faults.

I. INTRODUCTION

Humanoid robots have attracted a great attention of many researchers due to the suitability of the biped structure for tasks in the human environment. In this field, the ground reaction forces (GRF) [1] have crucial role in the humanoid stability and control. The zero moment point (ZMP) [2] and foot rotation indicator (FRI) [3] stability criteria both depend on the knowledge of the GRF which can be measured by sensors embedded in the robot feet of the humanoid robot as in [4-9] or estimated. The measured force may have faults and force sensors may have unreliable readings which would affect the control law adversely and thus cause the robot to lose its stability which may result in damaging the human being, the robot or parts of it, or the environment. A useful approach is to estimate the reaction forces to track the faults in the force sensor readings fixed in the feet sole of the robot.

The reaction force estimation for a standing robot using the joint torque information is reported in [4], where the motor current is used to calculate the torque, then the torque is mapped into force using the Jacobian. In [5], the total reaction force is estimated based on the inertial measurement unit (IMU) and joint encoders, with no indication to the contact points and their corresponding forces. In [6], the pseudo inverse is used to estimate the contact force distribution based on known ZMP, however, the pseudo inverse may fail and doesn't fulfill the reaction force constraints. The above studies were for controlling without using force sensors, however, force estimation can be used for sensor fault detection as in this paper.

Sensor faults [7] will inevitably affect the control laws adversely. Thus it is a necessity to design a fault detection and isolation (FDI) scheme to monitor and estimate the faults so that the controller can compensate for them in real time. The FDI methods are either model-free or model-based methods. The model-free FDI methods [8, 9] depend on the system features without using the system model, however, they don't reconstruct the faulty signals. The model-based FDI methods [10] utilize the system dynamic model that is operating in parallel to the real system and utilizing the same input. Their outputs are used to form the residual [10-16]. In the model-based methods, the faulty signal can be reconstructed however, its performance depends on the quality of the model [17]. Furthermore, obtaining the dynamic model for complicated systems such as the humanoid robot is a challenge due to the many degrees of freedom, coupling effects, nonlinear dynamics, and parameter and environment uncertainty [18]. An alternative method is to use software sensors or virtual sensors [19] based on simple mathematical models, this increase the measurement redundancy and reliability in the system. And they can be used for the fault detection and signal recovery as in this paper.

In this paper, a novel method for feet force sensors fault detection and tracking, and signal recovery based on virtual force sensor (VFS) is proposed. It is applicable on three types of faults: constant offset, drift and frozen output at constant value regardless the true value of the force. The VFS is designed to track the output of the force sensors assembled at

the foot sole of the robot leg while the foot is in full contact and assuming slow walking robot. The VFS has two functional steps: the first step requires the estimated center of mass (CoM) position which is reported by the author in [20], the leg kinematics, and the IMU readings to estimate the total reaction force and its location at each leg. In the second step, the total force is distributed optimally on the contact points for each leg by forming a constraint model and solving it using the quadratic programming optimization method. The number of the contact points and their locations are known in the foot frame. For the fault detection and recovery, a state space model for the measured forces from the sensors is formed including the fault, then the output of the VS is used to detect the faults and recover the true force for all the contact points simultaneously by using Kalman filter and threshold.

The rest of the paper is organized as follows: Section 2 describes the virtual force sensor design. Section 3 shows the threshold selection. Section 4 introduces the simulation platform and presents the simulation results. The paper conclusion is in Section 5.

II. VIRTUAL FORCE SENSOR VFS

The VFS model structure consists of two models: the linear inverted pendulum model (LIPM) and the Newton-Euler model. The LIPM function is to estimate the CoM position, while the Newton-Euler model is to estimate the ground reaction forces.

a) CoM position estimation

The CoM position c and the ZMP error p^{err} are estimated as in Fig. 1 by utilizing the actual Zero Moment Point ZMP trajectory p^{act} into the LIPM as presented by the author in [20]. The LIMP dynamics model in Eq(1) is used in its discrete form (Eq(2)) to form the process model in Kalman filter. The measurement model is the acceleration a_{com} measured from the IMU fixed the CoM. The error in the ZMP is modeled as the difference between the reference ZMP p^{ref} and p^{act} as in Eq(4). Then this error is decomposed into acceleration error and position error Δc based on the disturbance observer as in [21, 22]. The final corrected estimated Com position \hat{c} is obtained as in Eq(5).

$$\ddot{c} = \frac{g}{z_c} (c - p^{act}) \quad (1)$$

$$\ddot{c}_k = T \frac{g}{z_c} (\dot{c}_{k-1} - \dot{p}_{k-1}^{act}) + \ddot{c}_{k-1} \quad (2)$$

where

$$\dot{p}_{k-1}^{act} = \frac{p_k^{act} - p_{k-1}^{act}}{T} \quad (3)$$

and

$$p_k^{err} = p_k^{ref} - p_k^{act} \quad (4)$$

$$\hat{c}_k = c_k + \Delta c_k \quad (5)$$

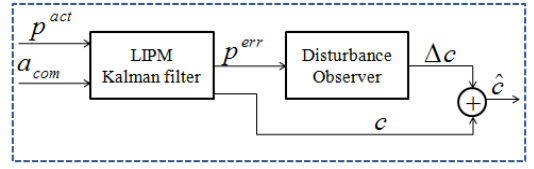


Fig. 1: CoM position estimation

where z_c is the constant height of CoM, g is the gravity acceleration (9.806 m/s^2), T is the sampling time, c is the CoM position, and k is the time instant.

b) Reaction force estimation

The force magnitude on each leg depends on the location of \hat{c} . Assuming the slow walking and neglecting the swinging leg and arms inertial forces, the forces for each leg are calculated using

$$F_L = \frac{1}{2}(1 + \alpha)m\ddot{c}, \quad F_R = \frac{1}{2}(1 - \alpha)m\ddot{c} \quad (6)$$

Where

$$\alpha = \frac{\hat{c}_y}{c_{y,max}}, \quad -1 \leq \alpha \leq 1 \quad (7)$$

$$\left\{ \begin{array}{l} \alpha = 0 \quad F_L = F_R \\ \alpha > 0 \quad F_L > F_R \\ \alpha < 0 \quad F_L < F_R \end{array} \right\}, \quad (8)$$

And

$$\left\{ \begin{array}{l} \alpha = 1 \quad \text{if } \alpha > 1 \\ \alpha = -1 \quad \text{if } \alpha < -1 \end{array} \right\} \quad (9)$$

where, m is the robot mass, F_R and F_L , are the reaction forces at the right and left feet respectively. \hat{c}_y is the estimated position of the CoM in the y-direction (lateral) and $c_{y,max}$ is the maximum of c_y obtained from the desired value or by recording the maximum of \hat{c}_y when the single support phase starts.

The robot is considered rotationally stable, so the angular momentum rate of change is zero [23]. For fully contact and not rotating foot, the moment from the CoM around the origin of the foot is

$$M_R = f_R \times F_R, \quad M_L = f_L \times F_L \quad (10)$$

Where f_R and f_L are the position of the right and left feet frame-origins respectively from the CoM, they are known by using the forward kinematics for each leg. M_R and M_L are moments at the right and left feet frame-origins respectively.

By the knowledge of the forces and moments, the location of the force d in the horizontal plane from the CoM point as projected on the horizontal plane is calculated for each foot using

$$d_i = [-M_{y,i} / F_{z,i} \quad M_{x,i} / F_{z,i}]^T \quad (11)$$

Where the subscripts $x, y,$ and z refer to the $x-, y-$ and $z-$ components respectively. The subscript i refers to right or left leg.

The Center of Pressure point CoP as expressed from the foot frame origin is

$$CoP_i = [d_i(1) - f_i(1) \quad d_i(2) - f_i(2)]^T, \quad (12)$$

which is constraint to the support polygon dimensions.

c) Distributed contact-point force estimation

The computed forces F_R and F_L are distributed among the contact points of the right and left feet respectively. The force distribution depends on the number of contact points n_R and n_L and their locations for the right and left feet respectively. For simplicity, an equal number of contact points for both feet is considered, i.e. $n_R = n_L = n$. The CoP coordinates (x_{CoP} and y_{CoP}) with F_z can be written in a matrix form as

$$CN = d \quad (13)$$

where

$$C = \begin{bmatrix} x_{R,1} & \cdots & x_{R,n} & 0_{1 \times n} \\ y_{R,1} & \cdots & y_{R,n} & 0_{1 \times n} \\ 0_{1 \times n} & x_{L,1} & \cdots & x_{L,n} \\ 0_{1 \times n} & y_{L,1} & \cdots & y_{L,n} \end{bmatrix}, N = \begin{bmatrix} F_{z,1} \\ \vdots \\ F_{z,2n} \end{bmatrix}, \quad (14)$$

and

$$d = \begin{bmatrix} F_{z,R} x_{CoP,R} \\ F_{z,R} y_{CoP,R} \\ F_{z,L} x_{CoP,L} \\ F_{z,L} y_{CoP,L} \end{bmatrix} \quad (15)$$

With the constraints

$$\left. \begin{aligned} F_{z,R} &= \sum_{l=1}^{n_R} F_{z,l} \\ F_{z,L} &= \sum_{l=2n-n_R}^{2n} F_{z,l} \end{aligned} \right\}, F_{z,l} \geq 0 \quad (16)$$

where x_k, y_k are the $x-$ and $y-$ positions of the contact point l that has the $z-$ reaction force $F_{z,l}$, $0_{m \times n}$ is a zero matrix of dimension $m \times n$.

Since C is a nonsquare matrix, the pseudo inverse of it can be calculated and then find N . However, the inverse suffers from singularities and doesn't fulfill the constraints in Eq (16). While the optimal solution is still required, the constraints in Eq(16) are written in matrix form as

$$A_{eq} N \leq b_{eq}, \quad (17)$$

$$AN = b \quad (18)$$

where

$$A_{eq} = -I_{2n}, b_{eq} = 0_{2n \times 1}, \quad (19)$$

$$A = \begin{bmatrix} 1_n & 0_{1 \times n} \\ 0_{1 \times n} & 1_n \end{bmatrix}, \text{ and } b = \begin{bmatrix} F_{z,R} \\ F_{z,L} \end{bmatrix} \quad (20)$$

where 1_m is row vector of one's of dimension m , and I_m is the identity matrix of dimension m .

For the optimal solution of the vector N , the cost function J is defined as

$$J = (CN - b)^T (CN - b) + N^T W N, \quad (21)$$

where W is a diagonal weighting matrix.

The quadratic programming optimization method is used to solve the least square problem subjected to the constraints in Eq(17) and Eq(18) as in

$$\begin{aligned} \hat{N} &= \min_N J \\ \text{s.t. } & A_{eq} N \leq b_{eq} \\ & AN = b \end{aligned} \quad (22)$$

After some mathematical manipulation, Eq(22) is written to be used directly in the quadratic programming method as

$$\begin{aligned} \hat{N} &= \min_N N^T (C^T C + W) N - d^T C N \\ \text{s.t. } & A_{eq} N \leq b_{eq} \\ & AN = b \end{aligned} \quad (23)$$

d) Fault detection and signal recovery

The output of the VFS \hat{N} in Eq(23) is used to estimate the faults in the force sensors and recover the faulty signal. The output of the sensors N_{meas} is modeled as the addition of the true force vector N , and the fault vector η as in

$$N_{meas} = N + \eta + v_N \quad (24)$$

And the fault is assumed to be a zero mean Gaussian noise $v_{\eta,k}$ driven as

$$\eta_k = \eta_{k-1} + v_{\eta,k} \quad (25)$$

then the true force vector is modeled as

$$N_k = N_{meas,k} - \eta_{k-1} - v_{\eta,k} \quad (26)$$

The Gaussian noises v_N and v_η are considered to be independent. For n contact points, Eq(25) and Eq(26) can be written in linear state space model as

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_k + w_k \\ Y_k &= Cx_k + v_k \end{aligned} \quad (27)$$

Where

$$x = \begin{bmatrix} \eta^T & N^T \end{bmatrix}^T, \quad (28)$$

$$u = \begin{bmatrix} 0_{1 \times n} & N_{meas}^T \end{bmatrix}^T, \quad (29)$$

$$A = \begin{bmatrix} I_{n \times n} & 0_{n \times n} \\ -I_{n \times n} & 0_{n \times n} \end{bmatrix}, B = \begin{bmatrix} 0_{n \times 2n} \\ 0_{n \times n} & I_{n \times n} \end{bmatrix}, \quad (30)$$

and

$$C = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \end{bmatrix} \quad (31)$$

w and v represent the process and the measurement noises respectively. They are considered Gaussian with covariance's Q and R as

$$\begin{aligned} w &\sim N(0, Q) \\ v &\sim N(0, R) \end{aligned} \quad (32)$$

Kalman filter is useful tool when the noise is contaminated in the system. It used to estimate the states in Eq (27), the estimated vector \hat{N} serves as the measurement vector in the correction part of Kalman filter.

For more stable fault detection due to the oscillations in the estimated η , it is filtered using low pas filter then used in the threshold in the next section.

III. THRESHOLD SELECTION

The threshold tr value is selected based on value of the filtered η for sensor j in the free fault case. Then the value of η determines whether a fault exists or not based on the selected tr as in

$$\begin{cases} |\eta_j| \leq tr_j, & \text{no fault}=0 \\ |\eta_j| > tr_j, & \text{fault}=1 \end{cases} \quad (33)$$

IV. SIMULATION RESULTS

The simulations are carried on 12 degrees of freedom (DOF) biped model. It consists of two 6-DOF legs and a trunk connecting them. Three joint axes are positioned at the hip, two joints are at the ankle and one at the knee (Fig. 2). The numerical values of the parameters (Table I) are taken to match our experimental humanoid robot SURALP (Sabanci University Robotics Research Laboratory Platform) [24]. The details of contact modeling and simulation algorithm are in [25]. The coordinate frames are shown in Fig. 3. All the measurements and calculation are in the world frame. The transformation is done using the rotational matrix obtained by the author in [26]. Kalman filter parameters are listed in Table II. The value $W = 0.01I_8$. The results shown here are for the right leg of the robot.

The output of the VFS compared to the true one is shown in Fig. 4. Since the VFS depends on the acceleration of the CoM, its output is bounded and doesn't suffer from the peeks as in the model. In the same context, the effect of the impact forces on its output is much less. In the simulation, three types of sensor faults are tested: offset, drift, and frozen output. These faults are added to the sensor output (the force). For the frozen output case, the sensor output is considered to have a constant value of zero. The considered fault for each force is shown in Fig. 5. For the force sensor at contact point 1 its output is considered 0, for the force sensor at contact point 2 its output is shifted by positive 100N, for the force sensor at contact point 3 its output has a time drift of slope 15, and the force sensor at contact point 4 its output has a negative shift of 100N. The estimated fault is shown in the same figure, because of high noise amplitude, the estimated fault is filtered using low pass filter and then used for the fault detection. The oscillations in the estimated fault are due to the error between the VFS output and the model. The estimated fault η (solid line) tracks the true fault (dashed line) as clear in Fig. 5. The recovered signal after removing the fault is depicted in Fig. 6. Recovering the signal in case of the offset and drift faults uses both the VFS output and the faulty signal which increases the accuracy of the estimation. For the frozen output, the estimation depends only on the VFS output. The fault detection is shown in Fig. 7 where the zero indicates no faults and the one indicates the fault occurrence. All the faults occur at the time instant 5, however their detection depends on the value of the threshold. In the first two seconds the threshold may indicate the fault existence while there is no fault. This is due the initialization of the algorithm. Different thresholds are applied, which explains the false detection and miss detection of the faults. Even the faults are applied suddenly the algorithm still able to track them and reconstruct the faulty signal.

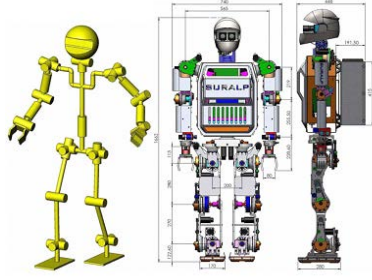


Fig. 2: The kinematic arrangement of SURALP

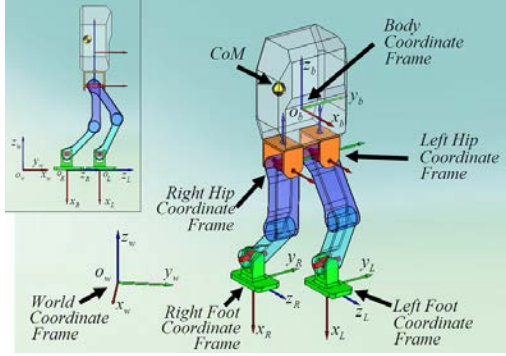


Fig. 3: Coordinate systems. o_w and o_b stand for the origins of the world and body coordinate frames, respectively. The foot coordinate frames are fixed to the foot soles.

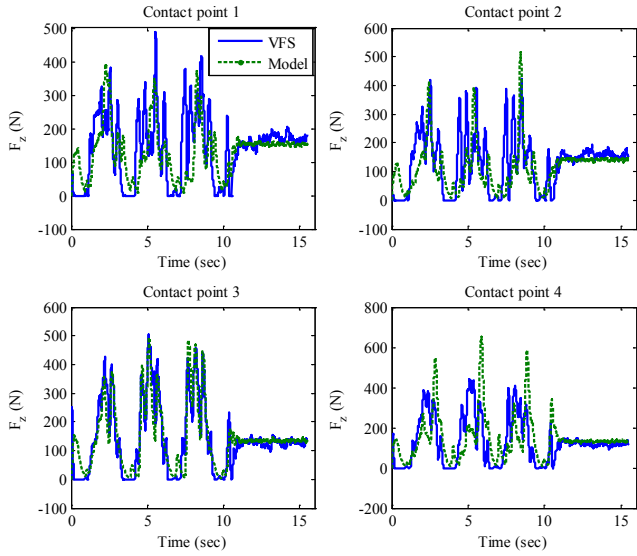


Fig. 4: The virtual force sensor output and the model force output

Table I: Robot Parameters

Parameter	Value
Upper leg length	280mm
Lower leg length	270mm
Sole-ankle distance	124mm
Foot dimensions	240mm×150mm
Upper arm length	219mm
Lower arm length	255mm
Robot weight	114 kg

Table II: simulation parameters

Parameter	value
Q	10^{-10}
R	10^{-4}
T	0.001

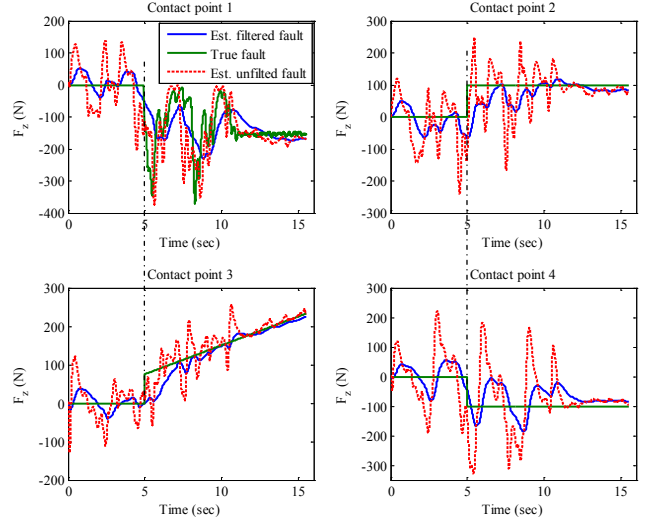


Fig. 5: The added fault (true), estimated filtered fault and the estimated unfiltered fault

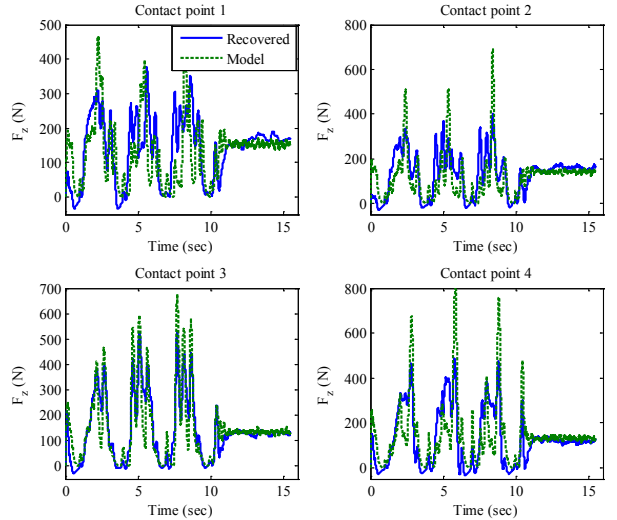


Fig. 6: The recovered signal (solid line) and the actual one (dotted line)

V. CONCLUSION AND FUTURE WORK

A method of detecting the force faults and reconstructing the faulty signal for a walking biped is proposed. It is based on designing the virtual force sensor which uses the CoM position and quadratic optimization method to estimate the distributed forces at the contact points of the feet. These forces are used as bases for the fault detection and faulty signal recovery. The algorithm uses both information from the sensor and the VFS to increase the accuracy of the estimation. The simulations validate the proposed method and motivate us to implement it into our humanoid robot SURALP in future.

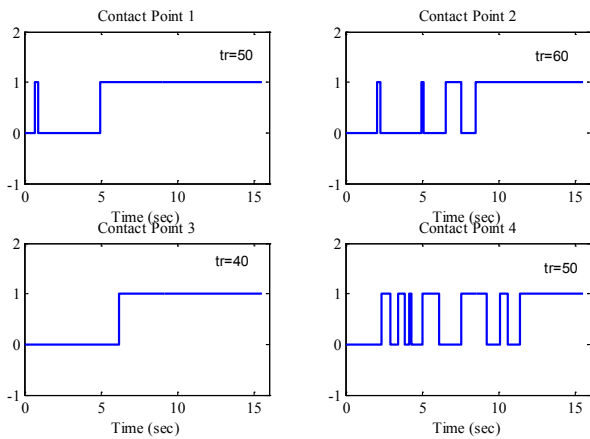


Fig. 7: Fault detection

REFERENCES

- [1] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. Center of pressure-zero moment point," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 34, pp. 630-637, 2004.
- [2] M. Vukobratović and B. Borovac, "Zero-moment point -thirty five years of its life," *International Journal of Humanoid Robotics (IJHR)*, vol. 01, pp. 157-173, 2004.
- [3] A. Goswami, "Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point," *The International Journal of Robotics Research*, vol. 18, pp. 523-533, June 1, 1999.
- [4] I. Hur, Y. Matsuki, N. Tomokuni, J. Huang, and T. Yabuta, "Standing Stability of Surfing Robot without Force Sensor," in the 25th Annual Conference of the Robotics Society of Japan, 3H11, 2007.
- [5] B. J. Stephens and C. G. Atkeson, "Dynamic Balance Force Control for compliant humanoid robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010 pp. 1248-1255.
- [6] H. Sang-Ho, "Compliant Terrain Adaptation for Biped Humanoids Without Measuring Ground Surface and Contact Forces," *IEEE Transactions on Robotics*, vol. 25, pp. 171-178, 2009.
- [7] H. Alwi, C. Edwards, C. Pin Tan, and C. P. Tan, "Fault Tolerant Control and Fault Detection and Isolation Fault Detection and Fault-Tolerant Control Using Sliding Modes," Springer London, pp. 7-27.
- [8] C. Seungdeog, B. Akin, M. M. Rahimian, and H. A. Toliyat, "Implementation of a Fault-Diagnosis Algorithm for Induction Machines Based on Advanced Digital-Signal-Processing Techniques," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 937-948, 2011.
- [9] H. Vermold and M. Saeger, "Plausibility Checking of Sensor Signals for Vehicle Dynamics Control Systems," in 8th International Symposium on Advanced Vehicle Control AVEC Taiwan, 2006.
- [10] R. Isermann, "Model-based fault-detection and diagnosis – status and applications," *Annual Reviews in Control*, vol. 29, pp. 71-85, 2005.
- [11] R. J. Patton and J. Chen, "Observer-based fault detection and isolation: Robustness and applications," *Control Engineering Practice*, vol. 5, pp. 671-682, 1997.
- [12] A. Khedher, K. Benothman, D. Maquin, and M. Benrejeb, "State and sensor faults estimation via a proportional integral observer," in 6th International Multi-Conference on Systems, Signals and Devices, 2009. SSD '09, pp. 1-6.
- [13] X.-G. Yan and C. Edwards, "Sensor fault detection and isolation for nonlinear systems based on a sliding mode observer," *International Journal of Adaptive Control and Signal Processing*, vol. 21, pp. 657-673, 2007.
- [14] J. Gertler, "Fault detection and isolation using parity relations," *Control Engineering Practice*, vol. 5, pp. 653-661, 1997.
- [15] R. Patton and C. Lopez-toribio, "Soft computing approaches to fault diagnosis for dynamic systems: a survey," in *Proceedings of the 4th IFAC Symposium SAFEPROCESS-00*, 2000.
- [16] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part III: Process history based methods," *Computers & Chemical Engineering*, vol. 27, pp. 327-346, 2003.
- [17] S. A. Arogeti, W. Danwei, L. Chang Boon, and Y. Ming, "Fault detection and isolation in a mobile robot test-bed," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2009. AIM 2009, pp. 398-404.
- [18] U. Seven, T. Akbas, K. Fidan, and K. Erbatur, "Bipedal robot walking control on inclined planes by fuzzy reference trajectory modification," *Soft Computing*, pp. 1-18, 2012.
- [19] G. Heredia and A. Ollero, "Virtual Sensor for Failure Detection, Identification and Recovery in the Transition Phase of a Morphing Aircraft," *Sensors*, vol. 10, pp. 2188-2201, 2010.
- [20] I. Hashlamon and K. Erbatur, "Center of Mass States and Disturbance Estimation for a Walking Biped" in *IEEE International Conference on Mechatronics, ICM 2013, Vicenza (Italy)*. 2013, pp. 248-253.
- [21] T. Sato, S. Sakaino, and K. Ohnishi, "Parameter design for ZMP disturbance observer of biped robot," in 34th Annual Conference of IEEE on Industrial Electronics, 2008, pp. 1650-1655.
- [22] T. Sato and K. Ohnishi, "ZMP disturbance observer for walking stabilization of biped robot," in *IEEE International Workshop on Advanced Motion Control*, 2008, pp. 290-295.
- [23] A. Goswami and V. Kallem, "Rate of change of angular momentum and balance maintenance of biped robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 3785-3790.
- [24] K. Erbatur, U. Seven, E. Taskiran, O. Koca, M. Yilmaz, M. Unel, G. Kiziltas, A. Sabanovic, A. Onat, and Ieee, *SURALP: A New Full-Body Humanoid Robot Platform*. New York: Ieee, 2009.
- [25] K. Erbatur and A. Kawamura, "A new penalty based contact modeling and dynamics simulation method as applied to biped walking robots," in *Proc. 2003 FIRA World Congress*, Vienna, Austria, 2003.
- [26] I. Hashlamon and K. Erbatur, "Experimental verification of an orientation estimation technique for autonomous robotic platforms," Master theses, Sabanci University, Istanbul, 2010.