IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 15, 2021, accepted February 8, 2021, date of publication February 22, 2021, date of current version March 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3061041

Bacterial Programming Based Kinematic Chain Estimation of Construction Vehicle

ZSIGMOND PÉNTEK[1,2], **JÁNOS BOTZHEIM**[1], **(Senior Member, IEEE), AND ANDRÁS CZMERK**[1]

[1]Department of Mechatronics, Optics and Mechanical ngineering Informatics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, 1111 Budapest, Hungary
[2]Robert Bosch Ltd., 1103 Budapest, Hungary

Corresponding author: Zsigmond Péntek (pentek.zsigmond@mogi.bme.hu)

This work was supported in part by the National Research Development and Innovation Fund (TKP2020 Institution Excellence Subprogram) under Grant BME-IE-MIFM and in part by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram) under Grant BME-NC.

ABSTRACT Construction vehicle automation for high accuracy applications require information about the state of the machine, resulting in a fully sensitized system with precise kinematic parameters. Since the measurement of these parameters contains uncertainties, accurate measurement of them is an expensive task. Automatic calibration of link parameters makes the task of kinematic parameter determination easier. This study reports a method for forward kinematic chain estimation of an excavator by bacterial programming (BP) based on randomly placed inertial navigation systems (INS) per segments with microelectromechanical sensors (MEMS) within. MEMS INS with fusion techniques provide increasing accuracy with outstanding resilience against harsh environment in a rigid housing. With known robot kinematic the tool orientation estimation can be made more accurate also the path can be planned. The unknown model structure and parameters are established and identified by BP without any a priori or given information about the device according to Denavit-Hartenberg (DH) transformation conventions. Fundamentals of this approach are described in detail and shown on simulated measurement results.

INDEX TERMS Bacterial programming, inertial navigation system, Denavit-Hartenberg, MEMS, kinematics, extended Kalman filter.

I. INTRODUCTION

Heavy construction equipments become more integrated and connected with automatising. Nevertheless it is hard to have exact information about all components, connections and restrictions that are inevitable for control system by tool positioning and path planning. Two methods of pose estimation are mainly applied in practice, the nonvisual sensor-based and the vision-based methods. Vision-based tool center point (TCP) position estimation methods can directly analyze position as well as orientation information [1] , however these methods are predominantly applied in indoor localization topics [2]. Marker-based vision systems recognize mounted markers and estimate the pose by their geometric relations [3], [4], or project infrared LEDs and analyze the patterns [5]. Markerless pose estimations directly extract image features and estimate pose [6] based on single [7] or multi [8] camera systems. As the vision name implies, a camera system must be set up to detect the markers or features, which is a hardly solvable task if the range of motion is as big as by an excavator also the rate of information update is very limited and computationally intensive. By non-visual sensor-based pose estimation systems the most deployed methods are: Inertial Measurement Unit (IMU), Global Positioning System (GPS), Wireless Local Area Network (WLAN), Radio Frequency Identification (RFID), and Ultra-Wide Band (UWB). Most of these methods cannot provide even partial information about the orientations of the excavator's segments making them unsuitable for navigation tasks except the IMU/INS. Application of INS is obvious thanks to MEMS technologies. They are small, cheap, easy to mount, can withstand harsh environment and also can provide absolute orientation in Earth-based coordinate system. Vihonen *et al.* in [9] shows improved tool positioning results incorporating gyroscopes, accelerometers and a well-known dynamical motion model. It provides a setup for heavy-duty serial-link manipulators that combines forward kinematics with complementary and Kalman filtering. Pivarčiová *et al.* [10] examined the assumption that an

The associate editor coordinating the review of this manuscript and approving it for publication was Frederico Guimarães.

industrial robot's trajectory is controllable with an integrated INS. Both studies confirmed that path planning and control of robot systems are possible if kinematic characteristics are known. Next to Vihonen *et al.* [9] and Pivarčiová *et al.* [10] it is stated also in [11], [12], that an Extended Kalman Filter (EKF) can reach higher orientation estimation accuracy with incorporated dynamics and with well known forward kinematics. All these studies choose the conventional way and they determined the theoretical kinematic model, which was used in the filter. Even if it is possible to accurately measure geometric parameters, the resulting model will still contain inaccuracies. Calibration with fully assembled system can bypass joint and link compliance, thermal effects, and some sensor errors. Because conventional calibration methods are based on classical mathematical regression, they typically fit a non-linear parametric model, which must be predefined by user [13]. The definition of model can be challenging because robotic inaccuracies are attributed to several error sources, what can be divided into three categories: joint errors, kinematic errors and non-kinematic errors. Joint errors represent displacement related errors provided by sensors which are the differences between measured and real movements. Kinematic errors are related to the knowledge of the robot's kinematic model where a not accurately represented geometry results in constant inaccuracy. These differences between the nominal and actual geometric dimensions are caused by manufacturing and assembling tolerances. Non-kinematic errors can be attributed to mechanical characteristics like component stiffness, gearbox backlash or temperature caused elongations [14], [15].

Various calibration techniques were presented in the past decades, but many rely on temporarily added sensors or measurement frameworks [16]. Ji *et al.* [17] have used coordinate measuring machine to measure end-effector position to joint angles. Drouet *et al.* [18] have presented a method to compensate for the geometric and elastic errors of a six-degree of freedom medical robot. Interferometer and laser tracking systems were also used for manipulator end-effector position measurement [19], [20]. In [21] kinematic parameters are determined only from measured joint angles. A calibration method by Jia *et al.* is presented in [22] for space manipulator kinematic parameters estimation. Common point in these works are the goal to increase TCP position accuracy with given model and parameter optimization after external measurements. In opposite of them this study provides a different approach, where the system does not have any information about the model or parameters. External reference measurement tools are also unavailable, the system can only rely on the mounted INS per segment and it has to determine the transformation matrices, which could be used by path planning and TCP positioning. Ikizoglu *et al.* [23] adapted and compared a neural network (NN) with several system identification model on a tilted turn table. Their results showed the possibility of mounting parameter calibration and kinematic chain estimation with INS sensors. El-Gohary *et al.* [24] built up the kinematic chain of a SCARA robot with IMUs on every segment to track the state with an Unscented Kalman Filter (UKF).

An evolutionary algorithm parameter optimization attempt to DH based kinematic chain estimation with given structure was done in [25], where Arellano *et al.* applied genetic programming to determine rotation and translation parameters based on robot arm encoders. Beside their results in direction of artificial intelligence, the proposed solution with encoders cannot handle stiffness caused payload error.

As the above literature shows, it is hard to consider all the sources that contribute to positioning errors in a single kinematic identification, however, most of the errors are related to geometric parameters of linkages. In recent years, stochastic optimization methods have gained increasing attention in parameter optimization. The most popular techniques are evolutionary computation, simulated annealing algorithms and swarm intelligence based techniques. Since these methods do not require gradient information, they are well suited for non-smooth or discontinuous optimization tasks occurring in nonlinear systems [26]. In this work a BP based kinematic chain estimation is proposed from INS based EKF angles. The model building approach adopted by BP is different from classical mathematical regression, because the relation between system input to output is not given but developed and derived in symbolic form. First of all this technique is not limited to a predefined model structure, therefore has the potential to generate an accurate calibration model. Secondly, through gravity based INS orientation estimation, it is insensitive against elastic deflections from external payloads. In BP the Denavit and Hartenberg (DH) conventions are used [27] to create the building blocks.

MEMS within INS are compact, lightweight and cost effective thus offering a cheap solution for mass production and are easy to integrate into various applications. An inertial system consisting of tri-axial accelerometer, gyroscope and magnetometer is commonly used for orientation estimation (roll - ϕ, pitch - θ, yaw - ψ angle). Fusion algorithms like Complementary Filter (CF), Gradient Descent (GD) and EKF can prevent many sensor errors and provide more accurate orientation results [28]. With calculated EKF angles as proposed in section III-A from rigidly mounted INS and with determined kinematic chain, the estimated link accelerations can be compared with measured ones and deviations can function as a measure of model accuracy. Thus manipulator kinematic can be identified after dynamic measurements by bacterial programming. The aim of this paper is to proof on simulated scenario, that kinematic model can be automatically built up, relying only on EKF angles from INS with simulated sensor characteristics.

The paper is organized as follows. In Section II the DH based kinematic analysis of a construction vehicle is explained, what is a black box for BP and has to be found out. Section III presents MEMS with angle estimation and data preparation with input/output pairs for our proposed method, described in Section IV. This method builds up a DH model to minimize the error between measured and estimated output

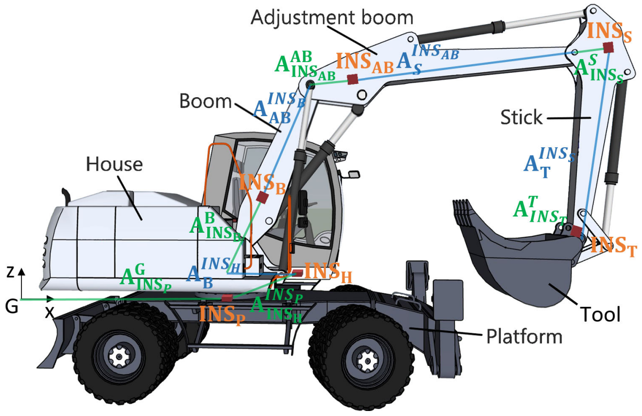


FIGURE 1. **Illustration of analyzed system with links, transformations and INS positions according to Table 1. Blue lines highlight transformations from INS to joints, green lines represent transformations from joints to INS. Figure was generated based on model [29].**

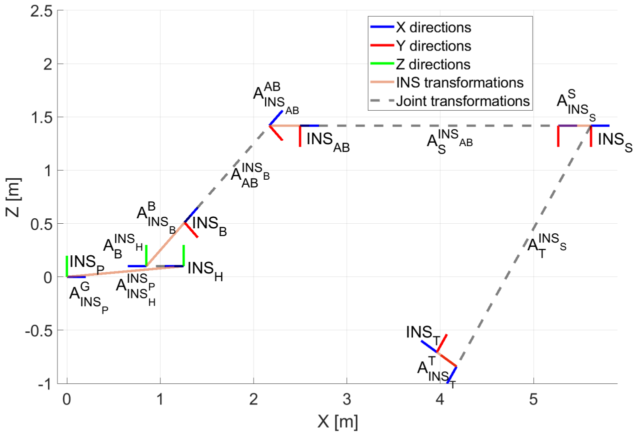


FIGURE 2. **Definition of the excavator pose. The pose of each segment can be calculated by the component's effective lengths and the angles between each component.**

values with substituted input values as Section V shows. Discussion and Conclusion are provided in Section VI and in Section VII.

II. KINEMATIC ANALYSIS OF A CONSTRUCTION VEHICLE

Kinematic chains are necessary to understand the vehicle's kinematic. A kinematic chain is a set of rigid bodies (links) coupled by relative motion constraining joints. The schematic of the excavator is defined in Fig. 1, where A_j^{j-1} is the homogeneous transformation matrix between the j and $j-1$ coordinate systems. All coordinate systems are right-handed Cartesian systems. The coordinate system of Earth is assumed as inertial one and without GPS tracking the platform's INS at the starting of motion is taken as reference position. Platform INS x-axis points horizontally forward. The y-axis is rectangular to axis x and points to the left side and z points upward. The angle convention of ϕ, θ and ψ is according to the right-handed system: clockwise is defined as positive. DH conventions are used to model the kinematics of the manipulator. In order to use a DH model, it is necessary to assign coordinate frames to each link and attached INS. A set of body and INS fixed coordinate frames is shown in Fig. 2. DH parameters are regarded as the products of two rotations and two translations as highlighted in Equation (1) after [27].

$$
\begin{aligned}
A_j^{j-1} &= Tran_{z_j,d_j} Rot_{z_j,\Theta_j} Tran_{x_j,a_j} Rot_{x_j,\alpha_j} \\
&= \begin{Bmatrix} \Theta_j \\ d_j \\ \alpha_j \\ a_j \end{Bmatrix} = \left[\begin{array}{c|c} R & T \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \\
&= \left[\begin{array}{ccc|c} c(\Theta_j) & -c(\alpha_j)s(\Theta_j) & s(\alpha_j)s(\Theta_j) & a_j c(\Theta_j) \\ s(\Theta_j) & c(\alpha_j)c(\Theta_j) & -s(\alpha_j)c(\Theta_j) & a_j s(\Theta_j) \\ 0 & s(\alpha_j) & c(\alpha_j) & d_j \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \qquad (1)
\end{aligned}
$$

where A_j^{j-1} is the homogeneous transformation matrix between the j and $j-1$ coordinate systems, c means cosine, s abbreviates sine function, R is the 3 × 3 direction cosine matrix and T is the 3 × 1 translation vector. Parameter d_j is the offset between z_j and z_{j-1} axes, Θ_j is the angle between x_j and x_{j-1}, a_j is the distance between z_j and z_{j-1} and α_j is the angle between z_j and z_{j-1}. The transformation between two frames can be expanded from link to link, to any frame in the system. To find the orientation and position of the tool INS frame relative to ground frame, the consecutive DH matrices must be multiplied from the ground to the end node as given in [27] and presented in Equation (2).

$$
A^G_{INS_T} = A^G_{INS_P} \cdot A^{INS_P}_{INS_H} \cdots A^T_{INS_T}, \qquad (2)
$$

The excavator is modeled by a rotational axis between the platform (P) and the house (H). The boom (A, AB) is in two parts, followed by a stick (S) and a tool (T). Capital letters in previous sentence serve as a segment ID notation. All segments are equipped with one randomly placed INS. As such, there are five revolute joints with the joint angles $\theta_{i=1..5}$, where i is the ID of the segment. Coordinate frames are attached to the joints and INS between two links, such that the first transformation is associated with the first joint, the second with the INS and the third is associated with the next link joint. Joints can be modeled as either revolute or prismatic to represent the relative movement between the links. The DH parameters for manipulator are shown in Table 1. Pose of the excavator can be described by the angle between each component (H, B, AB, S, and T) and by the 6 degree of freedom (DoF) coordinates of each joint (P - H, H - B, B-AB, AB - S, S - T). The pose of each excavator joint can be calculated using the angles and lengths of each component by forward kinematic. Therefore, determining the location and angle of each joint between each link is the primary goal.

In Table 1 the order of j^{th} link transformation is marked with an index. INS are mounted on every excavator link as highlighted in Fig. 1. They are used to measure inertial

TABLE 1. **Parameters used for DH parametrization of an excavator. There are solely rotational joints, going from the platform to the tool end node. *G* shortens Ground, *P* is Platform, *H* is House, *B* is Boom and *AB* is Adjustment Boom. Platform INS position by starting point is chosen as ground reference and is marked with $\{x_G, y_G, z_G\}$.**

Ind.	Transformation	a_j [m]	α_j [rad]	d_j [m]	Θ_j [rad]
1	$A^{G}_{INS_P}$	x_{INS_P}	0	0	0
2	$A^{INS_P}_{INS_H}$	-1.250	0	0	Θ_1
3	$A^{INS_H}_{B}$	0.400 0	0 $-\frac{\pi}{2}$	0 0	0 π
4	$A^{B}_{INS_B}$	0.580	0	0	Θ_2
5	$A^{INS_B}_{AB}$	1.285	0	0	0
6	$A^{AB}_{INS_{AB}}$	0.330	0	0	Θ_3
7	$A^{INS_{AB}}_{S}$	2.766	0	0	0
8	$A^{S}_{INS_S}$	0.350	0	0	Θ_4
9	$A^{INS_S}_{T}$	2.680	0	0	1.70
10	$A^{T}_{INS_T}$	0.100	0	0	Θ_5

and magnetic signals. In case of rotational and translational motion, a mounted INS unit provides orientation information after sensor fusion next to inertial values in body fixed coordinate system.

III. MICROELECTROMECHANICAL SENSORS

The integration of multiple sensor outputs in one device, called IMU/INS brings motion sensing systems to the level of accuracy required for the most demanding applications, such as indoor navigation and localization. INS consists of an accelerometer that measures experienced acceleration and a gyroscope which gives output about the observed rotational velocity. Together these two sensors measure 6DoF, which can relatively express the orientation of the unit. Apart from mechanical and robotic interpretation of DoF, the world of MEMS IMU uses a different terminology for ''degrees of freedom''. It does not represent real DoF, rather means adding up the dimensions that each type of sensor inside the IMU can detect. Extension of a 6DoF IMU with a triaxial magnetometer results in a 9DoF INS, which can predict absolute angles in Earth based coordinate system.

An accelerometer is a device which measures acceleration due to all forces acting on the device. Forces acting on a device include both the gravitational force due to the mass of the Earth as well as any inertial forces applied to the device. The two primary components of accelerations are, inertial and gravitational accelerations. Total acceleration measured by the device is the vector sum of these components $a_T = a_I + g$, where a_I is the inertial component and g is the gravitational component. Unless a body is completely motionless or moving with a constant velocity, there are inertial forces acting on it. If a sensor is motionless a time stable gravity based pitch and roll orientations can be calculated from acceleration signals, but additional inertial accelerations make this estimation inaccurate.

The angular velocity of a body can be described as the rate at which the object is rotating. A gyroscope can measure this angular velocity and with time integral, relative angles can be calculated. So derived angles are less noisy as from acceleration signals and independent from linear accelerations but due to sensor offset and temperature effects uncompensated errors cause constant drift in angle calculation.

Using a compass, Earth magnetic field can be exploited to determine the heading of an object to provide yaw angle reference for sensor fusion. Large metal objects and sources of electromagnetic interferences can distort this magnetic field, resulting in heading error so detection and correction of these distortions are crucial [30]. For this purpose Kalman filter can weight all information sources with knowledge about signal characteristics and system model. As proposed in [31] an EKF can estimate magnetic disturbance next to gyroscope bias and orientation.

To overcome these mentioned difficulties a sensor fusion method is needed which collects sensor measurements and after orientation calculation compares results with model predicted values from the previous state. Relative angular orientation can be determined by integrating angular rate sensor's signals. MEMS gyroscopes are accurate for angular velocity measurements but can be used to calculate orientation for a short time. A small offset error due to temperature or vibration interference inside the gyroscope will cause increasing integration errors [32]. Accelerometers measure the vector sum of linear accelerations and gravitational accelerations in sensor coordinates. Calculated orientations, using the angular rate sensor and magnetometer, can be used to express this vector sum in global coordinates. Gravitational acceleration components are in most fusion algorithms sensing dominant and provide information about inclination. This can be used to correct two of the three drifted orientation estimates (roll and pitch angles) from the gyroscope. Since angle around the yaw vertical axis from accelerometer signals cannot be measured, extension with magnetic sensing is needed. The magnetometer is sensitive to Earth's magnetic field and can be used as a reference to correct gyroscope integration caused angle drift around yaw axis.

A. SENSOR FUSION AND ANGLE ESTIMATION

IMU raw data was processed with a sensor-fusion algorithm using an iterated EKF approach based on [14], [31], however EKF is not the only solution for orientation calculation. As stated in [33], neural network could also be used to estimate orientation, but in this study EKF was selected, because it is easy to implement and with extra effort, as in [31], it can estimate magnetic disturbance caused errors. Also it provides Earth based absolute angles, which is needed for relative angle calculations. EKF is a nonlinear version of Kalman Filter (KF) using the first order Taylor expansion of nonlinear system equations around previous state. From Taylor expansion comes that state transition Equation (3) and observation models Equation (4) may be differentiable functions. The approximation is acceptable if the signal to noise ratio is low, or the non-linearity is small. Similar to the KF, there is a process function f and a measurement function

h that propagates the state $\hat{x}_{k|k-1}$ one time step through the function f. Predicted measurement $\hat{z}_k$ is found by propagating the predicted state $\hat{x}_{k|k-1}$ through the function h.

$$x_k = f(x_{k-1}) + w_{k-1} \tag{3}$$

$$z_k = h(x_k) + v_k, \tag{4}$$

where w_k and v_k are assumed to be uncorrelated white Gaussian noise with zero mean and covariances are given by Q_k as process and R_k as measurement noise respectively. R_k is filled up according to Table 2. Discrete time step is represented by k in the equations. Since the filter uses a state vector of estimates with corresponding covariances, matrices need to be updated to the new time step. In order to propagate an estimation $\hat{x} \in \mathbb{R}^n$, with mean μ_x and covariance matrix $\sum_x$, the function may be linearized.

The recursive prediction-measurement cycle is shown in Equations (5)–(12) [14]. EKF splits into one measurement update providing $\hat{x}_{k-1|k-1}, P_{k-1|k-1}$ and a time update yielding $\hat{x}_{k|k-1}, P_{k|k-1}$. A system is described by the state transition function f with process noise w_k, and a measurement model h with measurement noise vector v_k. The predicted state vector $\hat{x}_{k|k-1}$ in Equation (5), with its corresponding predicted error covariance matrix $P_{k,k-1}$ in Equation (6), is found by linearizing the f state transition function with its Jacobian around the estimated point as given in Equation (7).

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}) \tag{5}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{6}$$

$$F_k = \frac{\partial f(\hat{x}_{k-1|k-1})}{\partial \hat{x}_{k-1|k-1}} \tag{7}$$

The predicted measurement residual as given in Equation (8) is estimated using the predicted state vector. First order Jacobian linearization of measurement model h in Equation (12) around the predicted state allows the calculation of Kalman gain after Equation (9). Updated state estimate given by Equation (10) is updated using the Kalman gain in Equation (9) and its a priori estimate. For the next iteration the a posteriori error covariance matrix is updated according to Equation (11).

$$\hat{y}_k = z_k - h(\hat{x}_{k|k-1}, v_k) \tag{8}$$

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{9}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{y}_k \tag{10}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{11}$$

$$H_k = \frac{\partial h(\hat{x}_{k|k-1})}{\partial \hat{x}_{k|k-1}} \tag{12}$$

The state is defined in Equation (13) as a 12 × 1 vector [31]. λ_k^ϵ represents orientation errors. b_k^ϵ is the gyroscope's 3 × 1 bias change. A_k^ϵ is a 3 × 1 acceleration change in sensor frame, caused by linear accelerations. Its values must be subtracted from acceleration sensor measurements to accurately estimate gravity vector. M_k^ϵ is the 3 × 1 magnetic disturbance also in sensor frame caused by hard and soft iron magnetic interferences.

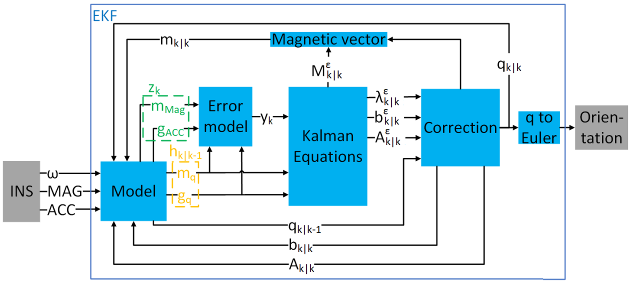


FIGURE 3. **Block diagram of error state estimation in three steps with EKF, with orientations represented in quaternions (*q*). First, model predicts gravity based accelerations from measurements (g_{ACC}) and from updated orientation (g_q) next to magnetic field predictions from error corrected magnetometer (m_{Mag}) and from updated orientation (m_q). Magnetic field strength is marked with *m*, and gravity with *g*. As second step, predicted error states ($\hat{x}^{\epsilon}_{k|k-1}$) are corrected with measured errors. As a last step, the estimated orientations, gyroscope offsets, linear accelerations and magnetic disturbances are updated with the estimated error states.**

$$x_k = \begin{bmatrix} \lambda_k^\epsilon & b_k^\epsilon & A_k^{\epsilon,S} & M_k^{\epsilon,S} \end{bmatrix}^T \tag{13}$$

Differently from previous segment naming convention, here S index means sensor based coordinates. A brief overview about model error estimating EKF is shown in Fig. 3.

The a priori angular velocity estimate is given as the gyroscope output (ω_k^S) corrected with the a posteriori gyroscope offset ($\hat{b}^S_{k-1|k-1}$). The a priori estimate of the orientation in quaternion ($\hat{q}_{k|k-1}$) is computed as a linear prediction from the a posteriori orientation ($\hat{q}_{k-1|k-1}$) and the a priori angular velocity estimate as given in Equation (14).

$$\hat{q}_{k|k-1} = \hat{q}_{k-1|k-1} quat\left(\frac{\omega_k^S - \hat{b}^S_{k-1|k-1}}{F_s}\right), \tag{14}$$

where $quat()$ means transformation from Euler angles to quaternion and F_s is the sampling frequency.

Measurement function (h) is defined as a 6 × 1 vector of estimated gravity ($\hat{g}_q^S$) and magnetic field ($\hat{m}_q^S$) from the a priori orientation. Measurements (z^S) from acceleration and magnetic sensors after compensations with linear acceleration and magnetic field disturbance are marked with $\hat{g}^S_{ACC}$, $\hat{m}^S_{Mag}$, respectively. Measurement residual in sensor coordinate frame can be calculated by these values after Equation (15).

$$\hat{y}_k = \left[z_k^S - h(\hat{x}_{k|k-1})\right] = \begin{bmatrix} \hat{g}^S_{ACC,k|k-1} - \hat{g}^S_{q,k|k-1} \\ \hat{m}^S_{Mag,k|k-1} - \hat{m}^S_{q,k|k-1} \end{bmatrix}$$
$$= \begin{bmatrix} \left(ACC_k^S + \hat{A}^S_{k|k-1}\right) - \left(\hat{R}_{k|k-1} g^G\right) \\ \left(MAG_k^S - m^S_{k|k-1}\right) - \left(\hat{R}_{k|k-1} m^G\right) \end{bmatrix} \tag{15}$$

The components of the a posteriori vector ($\hat{x}_{k|k}$) are corrected by the a posteriori error vector ($\hat{x}^{\epsilon}_{k|k}$). The a posteriori orientation matrix is computed by multiplying the a priori orientation matrix by a rotation matrix constructed from the negative of the a posteriori rotation vector corrections as shown in Equation (16).

$$\hat{q}_{k|k} = \hat{q}_{k|k-1} quat\left(-\hat{\lambda}^{\epsilon}_{k,k}\right), \tag{16}$$

The a posteriori estimate of the remaining terms can be computed by simple vector subtraction as presented in Equation (17).

$$\begin{bmatrix} \hat{b}_{k|k} \\ \hat{A}^{S}_{k|k} \\ \hat{M}^{S}_{k|k} \end{bmatrix} = \begin{bmatrix} \hat{b}_{k|k-1} \\ \hat{A}^{S}_{k|k-1} \\ \hat{M}^{S}_{k|k-1} \end{bmatrix} - \begin{bmatrix} \hat{b}^{\epsilon}_{k|k} \\ \hat{A}^{\epsilon,S}_{k|k} \\ \hat{M}^{\epsilon,S}_{k|k} \end{bmatrix} \tag{17}$$

B. DATA COLLECTION

For end-effector position calculation, link angles are necessary. In this work angles are calculated from 9DoF INS instead of encoders as described in Section III-A. With pressure sensor extension to 10DoF and knowledge about altitude, less restriction during BP has to be made as will be discussed later. Of course, the simplest and most accurate solution would be a pressure sensor with GPS extension in every INS with information about positions and relative INS height differences in the global coordinate system. On the other hand, GPS can be disturbed by signal blockage in a canyon or in a building also having varying inaccuracies [34], [35]. From this point of view it was excluded from simulation and only 9DoF INS devices were simulated. Simulink/Simscape was used to generate distorted sensor measurements while facing sequential movements. Signal sampling frequency (F_s) in simulation was set to 200Hz, which will be a feasible sampling rate also on the existing excavator. Link motion for model structure and extrinsic parameter calibration must satisfy several constraints. If there is no movement around all DoF, then some parameters are unobservable. From this consideration, excitation of a link must be persistent around all DoFs so, that measured inertial values remain within sensor's measurement ranges. In simulation typical sensor errors were included, to simulate real world disturbances and inaccuracies, whose characteristics are given in Table 2.

Identification of sensor series can be done with step by step actuator excitation. Gyroscope and acceleration noise can be measured before and after excitation. If INS is mounted on link after the excited one, then the measured signal noises will differ from references. A table like Table 3 can be determined where INS rank represents segment ID on the kinematic chain.

Because EKF determines orientation in ground reference coordinate system, but DH kinematic chain needs relative values, subtraction must be done according to Equation (18).

$$\phi_i^{i-1} = \phi_i^G - \phi_{i-1}^G \tag{18a}$$
$$\theta_i^{i-1} = \theta_i^G - \theta_{i-1}^G \tag{18b}$$
$$\psi_i^{i-1} = \psi_i^G - \psi_{i-1}^G, \tag{18c}$$

TABLE 2. Used sensor errors in simulation.

	Gyroscope
Measurement range	163 $\left[\frac{\circ}{s}\right]$
Angle random walk	9e-3 $\left[\frac{\circ}{\sqrt{s}}\right]$
Bias instability	1e-3 $\left[\frac{\circ}{s}\right]$
Rate random walk	1e-5 $\left[\frac{\circ}{s\sqrt{s}}\right]$
Quantization noise	5e-3 $\left[\frac{\circ}{s}\right]$
Filter bandwidth	60 $[Hz]$
	Accelerometer
Measurement range	4.2 $[g]$
Velocity random walk	2.2e-3 $\left[\frac{m}{s\sqrt{s}}\right]$
Bias instability	0 $\left[\frac{m}{s^2}\right]$
Acceleration random walk	9e-6 $\left[\frac{m}{s^2\sqrt{s}}\right]$
Quantization noise	1.3e-3 $\left[\frac{m}{s^2}\right]$
Filter bandwidth	60 $[Hz]$
	Magnetometer
Measurement range	1200 $[\mu T]$
White noise density	0.3 $\left[\frac{\mu T}{\sqrt{Hz}}\right]$
Bias Instability	70 $[nT]$
Quantization noise	0.3 $[\mu T]$
Filter bandwidth	20 $[Hz]$

TABLE 3. Link excitation and sensor measured signal deviation from reference, based on gyroscope and acceleration standard deviations. INS on link before excitation will stay motionless and comparable to references, thus becoming a 0 while moving INS becomes 1. After links are excited, dependencies of sensor's are determinable by summing table's columns. INS rank 1 represents the first INS on first link, while 2,3,... indicate further INS on kinematic chain's links.

Excitation	INS					
on	Platf	House	Boom	ABoom	Stick	Tool
House	0	1	1	1	1	1
Boom	0	0	1	1	1	1
Platf	1	1	1	1	1	1
Stick	0	0	0	0	1	1
ABoom	0	0	0	1	1	1
Tool	0	0	0	0	0	1
INS rank	1	2	3	4	5	6

where i is the ID of the segment after Table 3, ϕ_i^{i-1}, θ_i^{i-1} and ψ_i^{i-1} are the relative INS angles. First 3 × 3 matrix of DH components with relative angles between two INS according to Equations (18) represent directions, which can be used to validate model orientation compared to EKF calculated angles. Relative angular rates and accelerations can be derived from relative angles after Equation (19), which are also represented in global reference coordinate system just like in DH representation.

$$\epsilon_{i,x}^{i-1} = \dot{\omega}_{i,x}^{i-1} = \ddot{\phi}_i^{i-1} \tag{19a}$$
$$\epsilon_{i,y}^{i-1} = \dot{\omega}_{i,y}^{i-1} = \ddot{\theta}_i^{i-1} \tag{19b}$$
$$\epsilon_{i,z}^{i-1} = \dot{\omega}_{i,z}^{i-1} = \ddot{\psi}_i^{i-1}, \tag{19c}$$

where "$\cdot$" is the symbol of time derivative, ω is the relative angular velocity and ϵ is the relative angular acceleration. DH last column can be symbolically differentiated twice according to time to give reference based linear accelerations. Substituting calculated relative values from Equations (18) and (19) also serves as a measure of model accuracy. From linear components like $\omega^2\ R$ and ϵR, the INS mounting distance R from rotation joint can be determined.

Of course, longer measurements/simulations would contribute to better convergence of the applied BP algorithm and to reduce the effect of measurement noise but calculation time must be also kept in focus so a few second simulation was fed into BP to determine kinematic chain and its parameters.

IV. BACTERIAL PROGRAMMING

Machine learning is used to generate models of a system from data; these models should improve with more data, and they ideally generalize to scenarios beyond those observed in the training data.

In many tasks, the search space is exceedingly large and there may be multiple extrema so that gradient search algorithms yield sub-optimal results. Combining gradient search with Monte Carlo method may improve the quality of the solution, but this is extremely computational expensive. Evolutionary algorithms provide an effective alternative search strategy to find optimal solutions in a high-dimensional search space [36]–[38].

BP is an evolutionary algorithm that optimizes both the structure and parameters of an input–output map [39]. BP is a variant of Genetic Programming (GP) [40] which uses a tree structure similarly as GP does, however, the evolutionary operators applied in BP are inspired by Bacterial Evolutionary Algorithm (BEA) [41]. BP was successfully applied for designing B-spline Neural Networks [39], Hierarchical Fuzzy Systems [42], and for evolving a sensory-motor interconnection structure for realizing adaptive robot locomotion [43]. Another advantage for which BP was chosen is the faster solution convergence as stated in [39] and [42]. It outperforms the traditional GP method, because it needs less number of fitness evaluations to find a better solution.

The mapping discovered by BP is represented as a recursive function tree. For better understanding the BP terminology is shown in Fig. 4. The root of tree is the 4 × 4 DH homogeneous transformation matrix between start and end points. The tree contains several nodes, that can be branches or leaves. Branching points are mathematical operations, such as $\{+, -, \times, /\}$ and each branch may contain additional functions. Operation list was restricted to $\{\times\}$ in this given case because DH transformation is a series of matrix multiplications. The DH matrices are generated from leaves with 4 parameters according to DH convention. Leaves with DH parameters are filled through Equation (1) under the terminal node leaf name of DHP. Leaf parameter can be a constant or relative joint angle $(\phi_i^{i-1}, \theta_i^{i-1}, \psi_i^{i-1})$ between i and $i-1$ segments, derived from reference based EKF angles. A whole BP tree represents a set of DH transformations between segments with joint angles and constant inputs.

After an initial generation is built up with N_{ind} individuals and kinematic equations are derived from these individuals by DH representations, acceleration and orientation predictions are calculated and compared with INS measurements according to the cost function J_{RMSE}. The whole process from sensors to J_{RMSE} is presented in Fig. 5.

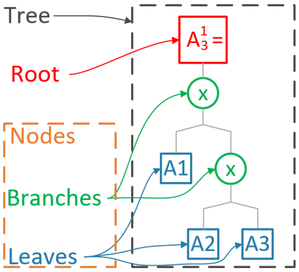


FIGURE 4. Representation of BP terminology. Each BP individual also called as tree represents a DH transformation between two points. The tree root is the 4 × 4 DH matrix. The tree contains several nodes, that can be branches with mathematical operations or leaves with DH parameters belonging to one transformation.

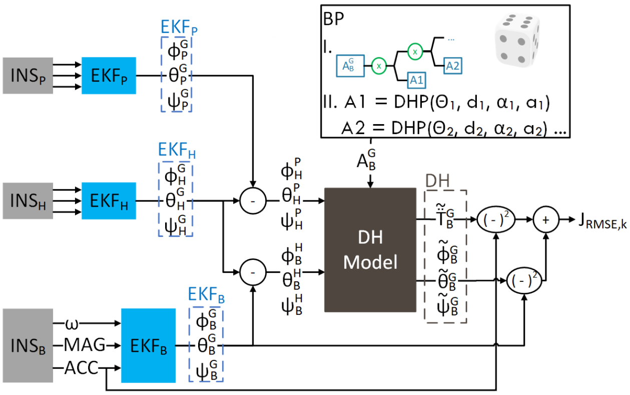


FIGURE 5. Process flow from INS sensors to J_{RMSE} calculation for a single k time step. Relative angles are calculated between EKF filters and substituted into BP determined DH model. Accelerations and orientations are calculated by DH model. They are then compared with sensor measured accelerations and EKF angles in ground coordinate system.

Between the generations individuals must change their DH structure and parameters to explore search space. The J_{RMSE} cost function of the determined models is evaluated based on the root mean squared error (RMSE) shown in Equation (20).

$$J_{RMSE} = \sqrt{\frac{\sum_{m=1}^{6}\sum_{k=1}^{n}(K_{m,k} - \widetilde{K}_{m,k})^2}{n}}, \tag{20}$$

where k is the time index of n measurements, m is the index of orientations and accelerations for the i^{th} segment, represented in ground coordinate systems as given in K in Equation (21). K contains the EKF angles and transformed acceleration sensor measurements for the i^{th} segment in the ground coordinate system. Similar to K, $\widetilde{K}$ contains the same parameters with the difference, that they are derived from the BP model and they are marked with ~ notation.

$$K_i = \left[\ddot{T}_{i,x}^G\ \ddot{T}_{i,y}^G\ \ddot{T}_{i,z}^G\ \phi_i^G\ \theta_i^G\ \psi_i^G\right] \tag{21}$$

In Equation (21) $\ddot{T}_i^G$ are the second order time derivatives of the translations between ground and the i^{th} segment's xyz directions, ϕ_i^G, θ_i^G ψ_i^G are the ground based segment angles.

Whole process is highlighted with an example between ground (G) and house (H) transformation (A_H^G). In this example BP created a DH model with two transformations with chosen parameters in Equation (22). α and Θ angles can be constants or joint angles from EKF as described in Equations (18), while translations are constants. After multiplication the 3×3 sub-matrix of transformation matrix represents the direction cosine matrix (DCM) and the 4^{th} column gives the translational vector, as highlighted in Equation (1).

$$A_H^G = A_P^G A_H^P = \begin{Bmatrix} \Theta_1 \\ d_1 \\ \alpha_1 \\ a_1 \end{Bmatrix} \begin{Bmatrix} \Theta_2 \\ d_2 \\ \alpha_2 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} \psi_H^P \\ 0 \\ 0 \\ -1 \end{Bmatrix} \begin{Bmatrix} 0 \\ 0 \\ \phi_H^P \\ 1.5 \end{Bmatrix}$$
$$= \begin{bmatrix} c\psi_H^P & -c\phi_H^P s\psi_H^P & s\phi_H^P s\psi_H^P & 0.5c\psi_H^P \\ s\psi_H^P & c\phi_H^P c\psi_H^P & -s\phi_H^P c\psi_H^P & 0.5s\psi_H^P \\ 0 & s\phi_H^P & c\phi_H^P & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

Ground reference system based orientations can be calculated from Equation (22) by Equations (23).

$$\hat{\phi}_H^G = tan^{-1}\left(\frac{R_{23}}{R_{33}}\right) = tan^{-1}\left(\frac{-s\phi_H^P c\psi_H^P}{c\phi_H^P}\right) \tag{23a}$$

$$\hat{\theta}_H^G = -sin^{-1}(R_{13}) = -s^{-1}\left(s\phi_H^P s\psi_H^P\right) \tag{23b}$$

$$\hat{\psi}_H^G = tan^{-1}\left(\frac{R_{12}}{R_{11}}\right) = tan^{-1}\left(\frac{-c\phi_H^P s\psi_H^P}{c\psi_H^P}\right) \tag{23c}$$

To calculate the linear accelerations from DH model, the Jacobian matrix (J_{Jac}) of the DH column with translational equations must be derived according to BP selected joint angle vector ($[\phi_H^P, \psi_H^P]^T$). J_{Jac} can be calculated by Equation (24). Note, that here we only calculate the linear velocities by Jacobian, angular velocities are omitted as long they are not used.

$$J_{Jac} = \begin{bmatrix} \frac{\partial T_H^P(x)}{\partial \phi_H^P} & \frac{\partial T_H^P(x)}{\partial \psi_H^P} \\ \frac{\partial T_H^P(y)}{\partial \phi_H^P} & \frac{\partial T_H^P(y)}{\partial \psi_H^P} \\ \frac{\partial T_H^P(z)}{\partial \phi_H^P} & \frac{\partial T_H^P(z)}{\partial \psi_H^P} \end{bmatrix} \tag{24}$$

Second time derivative of translational components of Equation (22) can be calculated by Equation (25).

$$\hat{\ddot{T}}_H^G = J_{Jac}\begin{bmatrix} \ddot{\phi}_H^P \\ \ddot{\psi}_H^P \end{bmatrix} + \dot{J}_{Jac}\begin{bmatrix} \dot{\phi}_H^P \\ \dot{\psi}_H^P \end{bmatrix}$$
$$= \frac{1}{2}\begin{bmatrix} -\ddot{\psi}_H^P s(\psi_H^P) - \dot{\psi}_H^{P,2} c(\psi_H^P) \\ \ddot{\psi}_H c(\psi_H^P) - \dot{\psi}_H^{P,2} s(\psi_H^P) \\ 0 \end{bmatrix}$$
$$= \frac{1}{2}\begin{bmatrix} -\epsilon_{H,z}^P s(\psi_H^P) - \omega_{H,z}^{P,2} c(\psi_H^P) \\ \epsilon_{H,z}^P c(\psi_H^P) - \omega_{H,z}^{P,2} s(\psi_H^P) \\ 0 \end{bmatrix} \tag{25}$$

TABLE 4. Bacterial programming parameter settings used by DH based forward kinematic chain estimation.

N_{ind}	100
N_{gen}	300
Mutation types	regrow, re-parametrization, leaf substitution
Function nodes	×
Terminal nodes	DHP(Θ,d,α,a)
Number of clones (N_{clones})	4
Mutation cycle (N_{cycles})	2
Transfer cycle ($N_{transfers}$)	50
Tree min depth (l_{min})	1
Tree max depth (l_{max})	3
Mutation min depth	1
Mutation max depth	2

Replacing time derivatives of relative joint angles according to Equations (19) leads to Equation (25). In order to compare sensor measured accelerations with accelerations from model, they must be rotated to the ground coordinate system with the inverse rotation matrix with given angles ($Rot^{-1}(\psi_H^G, \theta_H^G, \phi_H^G)$) as given in Equation (26). Because rotation is a square orthonormal matrix, its inverse is simply its transpose.

$$\ddot{T}_H^G = Rot^T(\psi_H^G, \theta_H^G, \phi_H^G) ACC_H$$
$$= \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} ACC_H \tag{26}$$

After all individuals within a generation are evaluated and sorted according to J_{RMSE}, a new generation is created. BP provides two steps for exploring search space. Bacterial mutation performs local optimization whilst the gene transfer allows the bacteria to directly transfer information to the other individuals in the population. This procedure is repeated until a convergence or stopping criterion is met, which is usually the number of generations (N_{gen}). Pseudocode of bacterial programming can be seen in Algorithm 1. The parameters of bacterial programming are presented in Table 4.

Algorithm 1 Process of Bacterial Programming

```
//Generating an initial population:
for $i \leftarrow 1$ to $N_{ind}$ do
    Generating tree structure of the $i$th individual
    Evaluating the $i$th individual
//Evolutionary loop:
for $i \leftarrow 1$ to $N_{gen}$ do
    Bacterial Mutation operation
    Ordering the population
    Gene Transfer operation
```

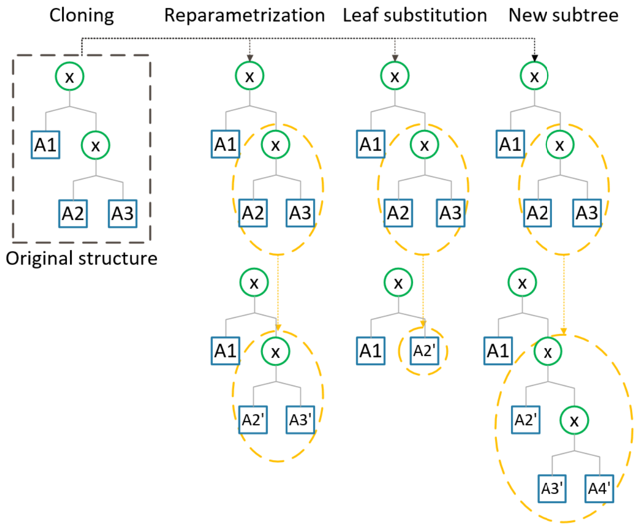


FIGURE 6. Three types of clone mutation during bacterial mutation after a node is selected. Tree reduction happens if a sub-tree is replaced by a leaf, expansion is possible through regrowing new sub-tree. The tree must be supervised and controlled in all cases according to min and max tree depth in Table 4.

Algorithm 1 can be repeated for segment after segment according to Table 3 and previous link's tree can be appended with further branches to model added link's kinematic. Without running BP for all links and INS, here the first 3 were investigated from Platform to Adjustment Boom. This relief is permissible because of the recursive behavior of the algorithm. Applying it on further links is just a matter of time because it needs to append the previously fixed tree with one sub-tree and optimize its parameters and structure so further modifications are not needed.

A. BACTERIAL MUTATION

Every candidate solution (so-called bacterium) is represented by a tree to be optimized. The length of each bacterium is initialized with a given depth. The length is maximized in each bacterial mutation and gene transfer operator to explore a search space with given complexity. Referring to prior knowledge, the size of search space is controlled by lower and upper boundaries.

The bacterial mutation operator creates clones and selects node in the parent tree and replaces the sub-tree at this node by a randomly generated sub-tree in all replica. As visualized in Fig. 6 new sub-tree can be regrown, reparameterized or replaced by a single leaf. Sub-tree length at the selected node remains the same in the case of reparametrization, but DH parameters are varied. In the case of shrinking substitution sub-tree can be replaced by a newly generated leaf, while on the other hand regrowing mutation can increase tree depth. In all cases tree depth must be supervised to avoid structure roughening or over-fitting.

Grow-mutation can result in a higher tree depth which must be checked and if needed sub-tree depth must be reduced. After clones are mutated, J_{RMSE} cost function is evaluated and clones are sorted. According to mutation cycle number only the best clone is picked always and sent to the next mutation cycle. Code segment about mutation is shown in Algorithm 2.

Algorithm 2 Bacterial Mutation Operation

```
for $i \leftarrow 1$ to $N_{ind}$ do
    Cloning $i^{th}$ individual into $N_{clones}$
    for $j \leftarrow 1$ to $N_{cycles}$ do
        Selecting mutation node
        for $k \leftarrow 2$ to $N_{clones}$ do
            Processing node mutation
        Evaluating mutated clones
        Selecting best individual from clones
        Best clone transfers mutated sub-tree
        to other clones
    Keeping best individual as new $i^{th}$ individual
    Deleting clones
```

Time complexity of bacterial mutation in one generation is $\mathcal{O}\{N_{ind} \cdot N_{clones} \cdot N_{cycles} \cdot l_{ind}\} \cdot \mathcal{O}\{t_{eval}\}$, where l_{ind} is the actual individual's length, and t_{eval} is the evaluation time of one stage deep tree. In one generation, there are N_{ind} individuals, and for each individual N_{clones} clones are used in the mutation.

B. GENE TRANSFER

Bacterial mutation is followed by the operation of gene transfer, which allows segment flow between two bacteria. The goal of the bacterial mutation is to explore the unknown searching space and bring in new beneficial information while gene transfer aims to preserve the incorporated features instead of introducing new features through passing them to other bacteria within the population. First, the population is divided into two sorted halves, a superior set and an inferior set, according to cost function J_{RMSE}. Then one bacterium is picked from both superior and inferior halves as the transfer candidates. A random sub-tree from the superior bacterium is transferred to the inferior bacterium and it overwrites a random sub-tree in the inferior bacterium. Gene transfer preserves existing information without incorporating new features. Larger trees imply expensive evaluation but they are not necessarily fitter than less complex trees, so a maximum depth constraint is imposed onto the trees in a population. Code segment about gene transfer is shown in Algorithm 3. The gene transfer process including the length modifications is shown in Fig. 7.

Time complexity of gene transfer in one generation is $\mathcal{O}\{N_{transfers} \cdot N_{ind} \cdot l_{ind}\} \cdot \mathcal{O}\{t_{eval}\}$, where l_{ind} is the actual individual's length, and t_{eval} is the evaluation time of one stage deep tree. For each transfer the N_{ind} individuals have to be sorted according to their J_{RMSE} results.

Algorithm 3 Gene Transfer Operation

```
for $i \leftarrow 1$ to $N_{transfers}$ do
    //Selecting a superior bacterium
    $Superior = i$
    //Selecting an inferior bacterium
    $Inferior = N_{ind} + 1 - i$
    Defining the sub-tree to be transferred
    from the Superior bacterium
    Defining the sub-tree to be overwritten
    in the Inferior bacterium
    Transferring and overwriting
    Evaluating the new Inferior bacterium
```

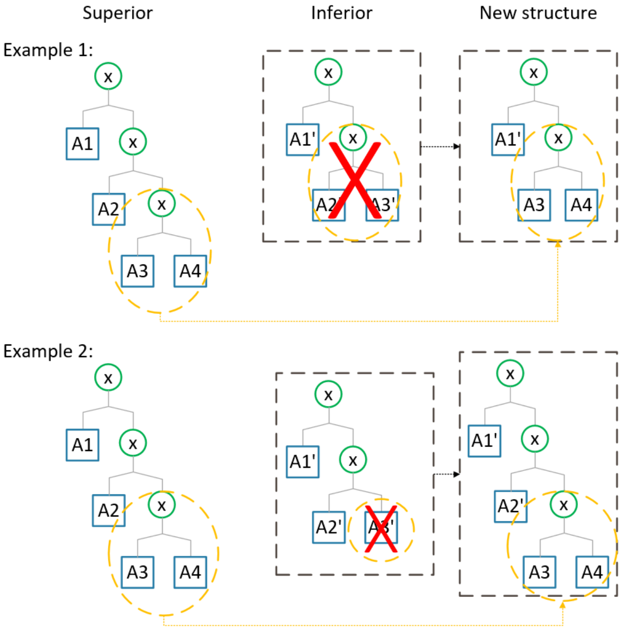


FIGURE 7. Example about gene transfer of bacterial programming, when superior is passing sub-tree section to inferior individual. Just like in mutation, tree dept must be controlled in gene transfer to match rules in Table 4.

V. RESULTS

The system is modeled in the local east north up (ENU) frame, with gravity of $g = 9.81m/s^2$. Accelerometers sense the gravity as well as the linear accelerations due to movement, where gyroscopes sense the angular velocities in sensor frame. After angles are calculated from sensor measurements with EKF filters, relative angular velocities and accelerations are computed according to Equations (19a)-(19c). When BP algorithm begins, the initial population with DH models are generated randomly without any a priori knowledge about the kinematics. From these models, orientations and accelerations in ground coordinate system are derived and the cost of each individual is evaluated by Equation (20). The next generation is produced through bacterial mutation and gene transfer. All tree modifier steps change one or more transformation steps in the DH model, after which the 4 × 4 matrix must be recreated with the direction cosine matrix and with the positions to derive the acceleration prediction for cost function. This process is continued iteratively until desired termination condition is reached. J_{RMSE} cost function was evaluated with ground coordinate system based x, y, and z accelerations and orientations using measured and BP estimated DH transformation derived values. After 300 generations of BP with 100 individuals, the results with determined transformation steps are summarized in Table 5.

TABLE 5. Original BP identified structure and parameters.

Ind.	Transformation	a_j [m]	α_j [rad]	d_j [m]	Θ_j [rad]
1	$A^{G}_{INS_P}$	x_G	0	0	0
2	$A^{INS_P}_{1}$	-2	-0.809	0	$\psi^{INS_P}_{INS_H}$
3	$A^{1}_{INS_H}$	0.738	0.810	0	0
4	$A^{INS_H}_{1}$	-0.400	-1.571	0	3.141
5	$A^{1}_{INS_B}$	0.575	0	0	$\theta^{INS_H}_{INS_B}$
6	$A^{INS_B}_{1}$	-1.307	0	0	0
7	A^{1}_{2}	2.593	0	0	0
8	$A^{2}_{INS_{AB}}$	0.340	0	0	$\theta^{INS_B}_{INS_{AB}}$

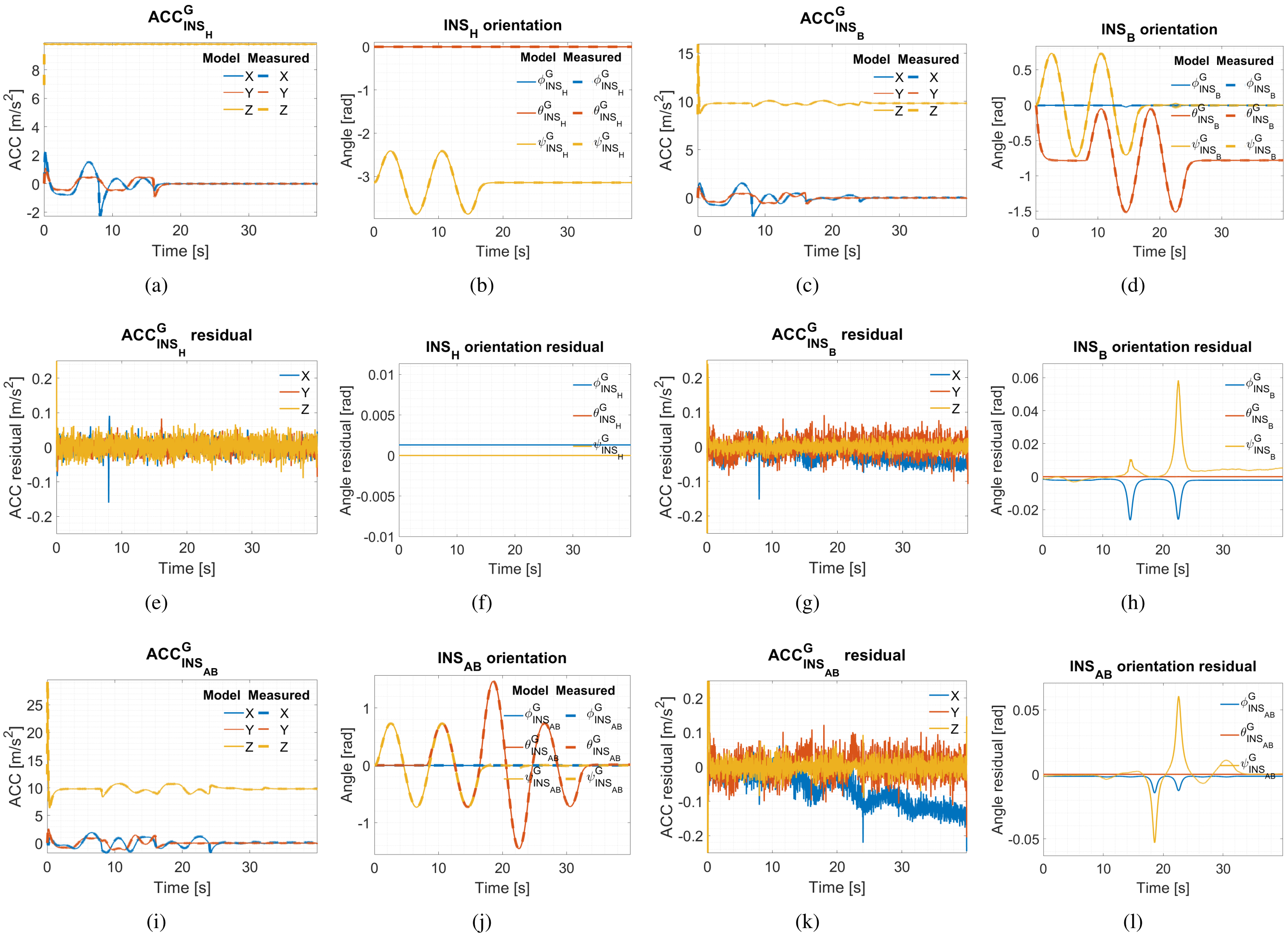
TABLE 6. Simplified and merged BP identified structure and parameters.

Ind.	Transformation	a_j [m]	α_j [rad]	d_j [m]	Θ_j [rad]
1	$A^{G}_{INS_P}$	x_G	0	0	0
2	$A^{INS_P}_{INS_H}$	-1.262	0.001	0	$\psi^{INS_P}_{INS_H}$
3	$A^{INS_H}_{B}$	-0.400	-1.571	0	3.141
4	$A^{B}_{INS_B}$	0.575	0	0	$\theta^{INS_H}_{INS_B}$
5	$A^{INS_B}_{AB}$	1.286	0	0	0
6	$A^{AB}_{INS_{AB}}$	0.340	0	0	$\theta^{INS_B}_{INS_{AB}}$

It is clearly visible, that between a few segments a simplification is possible because in BP determined transformations the translations are achieved in two steps instead of one. After result post-processing, these unnecessary splittings were simplified and merged together as shown in Table 6.

Merged, post-processed results from ground to INS_{AB} are given in Table 6 and are represented in equation form in Equation (28) to represent DH transformation matrices and multiplication orders. Next to estimated DH model from BP, the exact theoretical Equation (27) represents the solution with nominal geometric values from excavator data sheet.

$$\begin{aligned} Theoretic \quad A^{G}_{INS_{AB}} &= \begin{Bmatrix} 0 \\ 0 \\ 0 \\ x_G \end{Bmatrix} \begin{Bmatrix} \psi^{INS_P}_{INS_H} \\ 0 \\ 0 \\ -1.250 \end{Bmatrix} \begin{Bmatrix} 3.141 \\ 0 \\ -1.571 \\ -0.400 \end{Bmatrix} \\ &\quad \begin{Bmatrix} \theta^{INS_H}_{INS_B} \\ 0 \\ 0 \\ 0.580 \end{Bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1.285 \end{Bmatrix} \begin{Bmatrix} \theta^{INS_B}_{INS_{AB}} \\ 0 \\ 0 \\ 0.330 \end{Bmatrix} \qquad (27) \\ BP \quad A^{G}_{INS_{AB}} &= \begin{Bmatrix} 0 \\ 0 \\ 0 \\ x_G \end{Bmatrix} \begin{Bmatrix} \psi^{INS_P}_{INS_H} \\ 0 \\ 0.001 \\ -1.262 \end{Bmatrix} \begin{Bmatrix} 3.141 \\ 0 \\ -1.571 \\ -0.400 \end{Bmatrix} \end{aligned}$$



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

FIGURE 8. Accelerations from second order time derivative of BP determined DH transformation matrices last column are visualized on (a, e, i). Calculated Euler angles from DH first 3 × 3 direction cosine matrix are shown on (b,f,j) for the first three links. Model estimation errors are visualized on (c, d, g, h, k, l) in time in comparison to transformed INS measurements.

$$\begin{Bmatrix} \theta^{INS_H}_{INS_B} \\ 0 \\ 0 \\ 0.575 \end{Bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1.286 \end{Bmatrix} \begin{Bmatrix} \theta^{INS_B}_{INS_{AB}} \\ 0 \\ 0 \\ 0.340 \end{Bmatrix} \tag{28}$$

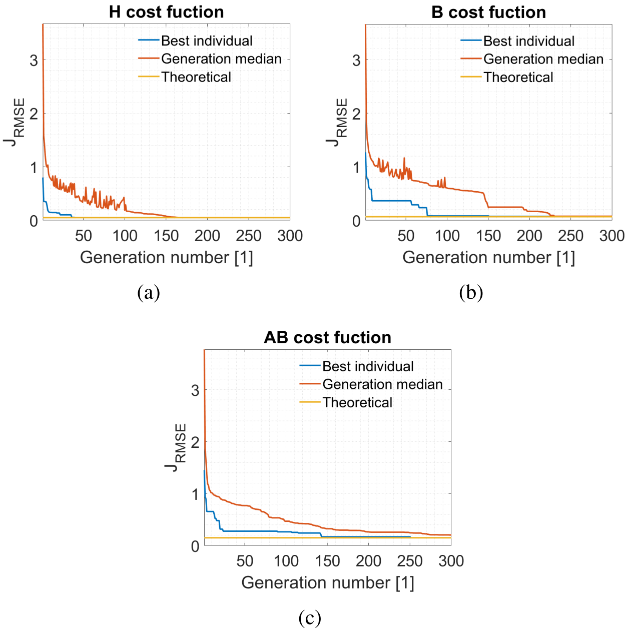
As it is clearly visible, the structure of the equations are identical, only small deviations in constants are observable causing some measurable error in cost functions. We believe, that letting BP to evolve through further generations, the best optimal parameters would be found with even less error. Orientation and acceleration characteristics of derived functions for first three links from given BP Equation (28) are evaluated with derived INS measurements after Equations (18) and (19). As given in Equation (21) these parameters are the mains, serving as a measure of model accuracy. Measured and calculated accelerations and orientations for the best individuals, found for the first 3 segments are highlighted as a function of time in Fig. 8 (a-d, i-j) with residual errors after subtracting measurements from predictions in Fig. 8 (e-h, k-l).

Spikes in Fig. 8 (h,l) ϕ are the cross-sensitivity with ψ angle in BP model, caused by the small miscalibrated ϕ angle in the $A^{INS_P}_{INS_H}$ and the poor π value representation between $A^{INS_H}_{B}$. If these errors were 0[rad], then the whole error would pop up in the ψ angle. Those small spikes in ψ residual plots in Fig. 8 (h, l) shows up within EKF because of uncompensated linear accelerations. Slowly diverging ACC_X in Fig. 8 (g, k) are caused by the drift of EKF calculated θ and ψ angle, that are used to rotate INS measured accelerations to ground coordinate system. Because DH models for INS_B and INS_{AB} do not depend on ψ angle between INS_H and INS_B, model estimated segment angles and accelerations do not drift as EKF.

Theoretical and BP identified DH model's J_{RMSE} cost function according to Equation (20) with acceleration and orientation errors are shown in Table 7. Because sensor measured accelerations must be rotated to the ground coordinate system with error loaded EKF angles and linear accelerations have influence on the EKF estimated angles despite of an adjusted higher covariance in system model, the best possible cost function values with nominal geometric values won't be zero. An INS overcoming higher linear accelerations through movements also can have higher EKF angle errors and with that a higher reachable cost function minimum. Calculating the relative errors of segment's cost functions, an average

TABLE 7. J_{RMSE} **cost function between calculated and measured accelerations and orientations for house, boom and adjustment boom INS. INS forward kinematics contains always the previous INS transformations with the additional transformation to actual measurement device position.**

J_{RMSE}	H	B	AB
BP Model	0.050	0.0699	0.1518
Theoretical	0.050	0.0686	0.1507



(a)

(b)

(c)

FIGURE 9. **Cost function of best individuals between generations with bacterial programming for the first three links also the converging median of the whole population.**

of 0.31% deviation is present in BP, compared to theoretical solutions.

The values of the cost functions for the best individuals in each generation are shown in Fig. 9. It is visible on all three links generation's median, how the whole population converges to best individual thanks to gene transfer, also by jumps how the best cost function changes if a better individual is found after bacterial mutation.

VI. DISCUSSION

In this study DH based kinematic model of a sensed excavator was determined by BP without any a priori knowledge or user intervention using only the mounted INS per segments. The implementation of this model in an EKF results in more accurate angle estimation through linear acceleration forecasting and compensation, providing also a tool positioning and path planning possibility for a control system. Positioning INS further away from rotational axis is beneficial for parameter estimation thanks to the occurring higher inertial accelerations but disadvantageous for non-model aided EKF orientation estimation, also gyroscope and accelerometer clipping should not be forgotten. Small distance will result in negligible inertial accelerations, that can disappear in sensor's noise level, making the mounting position estimation indeterminable. The whole kinematic chain estimation process is extremely sensitive on EKF angle estimation from INS measurements so all related errors must be avoided or reduced as much as possible. For this reason EKF process model includes gravitational and inertial acceleration components with enlarged model covariance to relieve inertial acceleration caused angle estimation errors. This setting will prioritize gyroscope and magnetometer measurements over acceleration sensors, meaning that magnetic field must be undisturbed from soft- and hard-iron effects during calibration data collection. Small angle errors were still present in dataset after EKF parameter adjustments, what can be seen in Table 7 through the nonzero J_{RMSE} after nominal parameter substitution into theoretical DH model. Another source of non-zero J_{RMSE} cost function in case of known settings is the distributed noise in derived relative angular velocities, in angular accelerations and in accelerometer measurements. Despite J_{RMSE} values will not be zero in any case, they are converging to analytical ones, which can be seen on generation based cost functions in Fig. 9. As illustrated, BP converges quickly in initial populations and slows down in later courses of evolution.

For first three links calculated accelerations from second order time derivative of BP determined DH transformation matrices, and Euler angles from first 3 × 3 direction cosine matrices are visualized compared to INS measured accelerations and EKF orientations. To better see time dependent deviations, errors between BP model and measured values are visualized in Fig. 8 (c, d, g, h, k, l). A 0.001[rad] ϕ angle error from DH model is visible in Fig. 8 (f), which will accumulate to afterward links. Radius inaccuracy caused inertial accelerations and angle error caused gravitational accelerations are showed in Fig. 8 (e,g,k). Another unavoidable J_{RMSE} error source is measurement noise, what is visible on residual plots in Fig. 8 (e-h,k-l). Comparing the found parameters of transformations with theoretical values the maximum length and angle errors are 1.2[cm] and 0.001[rad]. To increase the possible accuracy, the application of redundant sensor arrays or more accurate sensors with less noise could serve as an opportunity. In this case smaller inertial accelerations could also be detected and described in the model.

After DH model determination is ready, derived model can be put in EKF to compensate calculated linear accelerations, also EKF linear acceleration covariance can be reseted to have more trust in accelerometer determined ϕ and θ angles.

A drawback of the proposed method is the spreading relative angle error along links after EKF estimation. A second issue is the accumulating BP determined model parameter inaccuracy, causing unavoidable orientation and acceleration errors by INS on farther links. Thirdly, this method is efficient in rotational joint estimation, because for translational joints relative INS positions with known initial values would be necessary next to highly accurate sensors to integrate linear accelerations twice. To fully determine the kinematic chain of an open loop robotic manipulator the TCP orientation and geometric parameters must be known according to mounted INS, because without this additional knowledge or without a temporary mounted reference INS its relative position on tool is indeterminable.

VII. CONCLUSION

INS offers greater placement flexibility and less cost than absolute encoders or vision-based systems, nonetheless can provide precise measurement. Proposed method uses motion-based information with no a priori knowledge about the system kinematics to recover it. Performance of BP determined DH transformation matrix is evaluated on simulated INS data. Each link of excavator has assigned geometry and a mounted virtual INS with given sensor characteristics. Persistent test movement was simulated, resulted in data of acceleration, magnetic field strength and angular velocity, which was then used for EKF orientation calculation to derive relative motions in ground coordinate system. By proposed test sequence, dependencies of links and connections can be determined and arranged in a hierarchical tree. Relative motions can then be fed into BP determined DH model to calculate orientations and accelerations. Comparing the differences between measured and model calculated orientations and accelerations, Equation (20) provides quantitative measure to classify DH model accuracy represented as a BP individual. BP can mutate individuals for local search and it can also transfer genes from superior individuals to inferior, covering a bigger search space. Tree building with BP can be built up sequentially, meaning a simple sub-tree expansion for next link in previously determined tree. Doing so, DH based forward kinematic chain of excavator's first 3 links with mounted INS was determined. Results in the form of Equation (28) clearly indicate the applicability of used method in case of rotational joints. BP determined DH model can be used in EKF to compensate linear accelerations, what can cause angle errors in EKF gravity's based orientation estimation. Secondly determined DH model can be used for path planning and to calculate TCP orientation.

Advantages of using intelligent methods to determine DH transformation matrix are:

- They do not require a priori information about model structure and parameters
- Their applications are simple
- They don't get trapped in local minima as easily as classic regression methods

In case of a robot, restricted to fully rotational joints, the proposed algorithm also has an advantage that it does not require external equipments like cameras or markers for data collection, only the mounted INS on segments. MEMS with BP can be used to obtain not only geometric parameters but assembling inaccuracies without system preparation unlike other vision based motion systems.

As next step a measurement on real excavator with proposed sensor fusion algorithm in [32] is planned in a 9DoF setup. With the gained values, the here proposed BP based kinematic chain estimation could be rerun and evaluated. It will be explored how the aforementioned limitations and accumulating errors could be mitigated or eliminated.

ACKNOWLEDGMENT

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 Institution Excellence Subprogram, based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

János Botzheim was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

The authors want to particularly thank Tamás Fischl and Burkhard Kuhlmann for their generous help and advices.

REFERENCES

[1] M. M. Soltani, Z. Zhu, and A. Hammad, "Framework for location data fusion and pose estimation of excavators using stereo vision," *J. Comput. Civil Eng.*, vol. 32, no. 6, Nov. 2018, Art. no. 04018045.

[2] B. R. K. Mantha, C. C. Menassa, and V. R. Kamat, "Robotic data collection and simulation for evaluation of building retrofit performance," *Autom. Construct.*, vol. 92, pp. 88–102, Aug. 2018.

[3] C. Feng, V. R. Kamat, and H. Cai, "Camera marker networks for articulated machine pose estimation," *Autom. Construct.*, vol. 96, pp. 148–160, Dec. 2018.

[4] G. Nagymáté and R. M. Kiss, "Affordable gait analysis using augmented reality markers," *PLoS ONE*, vol. 14, no. 2, Feb. 2019, Art. no. e0212319.

[5] W. Yang, X. Zhang, H. Ma, and G.-M. Zhang, "Infrared LEDs-based pose estimation with underground camera model for boom-type roadheader in coal mining," *IEEE Access*, vol. 7, pp. 33698–33712, 2019.

[6] C.-J. Liang, K. M. Lundeen, W. McGee, C. C. Menassa, S. Lee, and V. R. Kamat, "A vision-based marker-less pose estimation system for articulated construction robots," *Autom. Construct.*, vol. 104, pp. 80–94, Aug. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580518311579

[7] B. Solvang, G. Sziebig, and P. Korondi, "Vision based robot programming," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, Apr. 2008, pp. 949–954.

[8] T. Pallos, G. Sziebig, P. Korondi, and B. Solvang, "Multiple-camera optical glyph tracking," *Adv. Mater. Res.*, vol. 222, pp. 367–371, 2011.

[9] J. Vihonen, J. Honkakorpi, J. Tuominen, J. Mattila, and A. Visa, "Linear accelerometers and rate gyros for rotary joint angle estimation of heavy-duty mobile manipulators using forward kinematic modeling," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 3, pp. 1765–1774, Jun. 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7437461

[10] E. Pivarčiová, P. Božek, Y. Turygin, I. Zajačko, A. Shchenyatsky, Š. Václav, M. Císar, and B. Gemela, "Analysis of control and correction options of mobile robot trajectory by an inertial navigation system," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 1, 2018, Art. no. 1729881418755165, doi: 10.1177/1729881418755165.

[11] E. Dorschky, M. Nitschke, A.-K. Seifer, A. J. van den Bogert, and B. M. Eskofier, "Estimation of gait kinematics and kinetics from inertial sensor data using optimal control of musculoskeletal models," *J. Biomech.*, vol. 95, Oct. 2019, Art. no. 109278. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021929019304841

[12] I. G. Fernandez, S. A. Ahmad, and C. Wada, "Inertial sensor-based instrumented cane for real-time walking cane kinematics estimation," *Sensors*, vol. 20, no. 17, p. 4675, Aug. 2020.

[13] Z. Pentek, T. Hiller, T. Liewald, B. Kuhlmann, and A. Czmerk, "IMU-based mounting parameter estimation on construction vehicles," in *Proc. DGON Inertial Sensors Syst. (ISS)*, Sep. 2017, pp. 1–14.

[14] G. Rigatos and K. Busawon, *Robotic Manipulators and Vehicles: Control, Estimation and Filtering*, vol. 152. New York, NY, USA: Springer, 2018.

[15] M. Barati, A. Khoogar, and M. Nasirian, "Estimation and calibration of robot link parameters with intelligent techniques," *Iranian J. Elect. Electron. Eng.*, vol. 7, pp. 225–234, Dec. 2011.

[16] P. Muller, M.-A. Begin, T. Schauer, and T. Seel, "Alignment-free, self-calibrating elbow angles measurement using inertial sensors," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 2, pp. 312–319, Mar. 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7782745

[17] J. Ji, L. Sun, and L. Yu, "A new pose measuring and kinematics calibrating method for manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 4925–4930.

[18] P. Drouet, S. Dubowsky, S. Zeghloul, and C. Mavroidis, "Compensation of geometric and elastic errors in large manipulators with an application to a high accuracy medical system," *Robotica*, vol. 20, no. 3, pp. 341–352, May 2002.

[19] L. Zhao, A. Joubair, P. Bigras, and I. A. Bonev, "Metrological evaluation of a novel medical robot and its kinematic calibration," *Int. J. Adv. Robotic Syst.*, vol. 12, no. 9, p. 126, Sep. 2015, doi: 10.5772/60881.

[20] P.-N. Le and H.-J. Kang, "A new hybrid calibration method for robot manipulators by combining model–based identification technique and a radial basis function–based error compensation," in *Intelligent Computing Methodologies*, D.-S. Huang, Z.-K. Huang, and A. Hussain, Eds. Cham, Switzerland: Springer, 2019, pp. 20–31.

[21] P. Marić and V. Potkonjak, "Geometrical parameter estimation for industrial manipulators using two-step estimation schemes," *J. Intell. Robotic Syst.*, vol. 24, no. 1, pp. 89–97, 1999.

[22] Q. Jia, W. Shao, G. Chen, Y. Wang, and L. Li, "A self-calibration method of lander manipulator for deep space exploration mission," in *Intelligent Robotics and Applications*. Cham, Switzerland: Springer, 2019, pp. 354–366.

[23] S. Ikizoglu and Y. Kamer, "Acceleration data correction of an inertial navigation unit using turntable test bed," in *Proc. World Congr. Elect. Eng. Comput. Syst. Sci.*, Barcelona, Spain, 2015, pp. 1–9, Paper 149.

[24] M. El-Gohary and J. McNames, "Human joint angle estimation with inertial sensors and validation with a robot arm," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 7, pp. 1759–1767, Jul. 2015.

[25] H. Velasco Arellano and M. Montes Rivera, "Forward kinematics for 2 DOF planar robot using linear genetic programming," *Res. Comput. Sci.*, vol. 148, no. 6, pp. 123–133, Dec. 2019.

[26] M. Sreedhar, S. A. N. Reddy, S. A. Chakra, T. S. Kumar, S. S. Reddy, and B. V. Kumar, "A review on advanced optimization algorithms in multidisciplinary applications," in *Recent Trends in Mechanical Engineering*. Singapore: Springer, 2020, pp. 745–755.

[27] R. S. Hartenberg and J. Denavit, "A kinematic notation for lower pair mechanisms based on matrices," *J. Appl. Mech.*, vol. 77, no. 2, pp. 215–221, 1955.

[28] B. Liu, M. Adams, and J. Ibanez-Guzman, "Multi-aided inertial navigation for ground vehicles in outdoor uneven environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 4703–4708.

[29] Jos. (2014). *Terex Excavator*. Accessed: Sep. 21, 2020. [Online]. Available: https://3dwarehouse.sketchup.com/model/459ad2317cae7cc2fd7f36b881b7abf3/Terex-excavator

[30] F. Wittmann, O. Lambercy, and R. Gassert, "Magnetometer-based drift correction during rest in IMU arm motion tracking," *Sensors*, vol. 19, no. 6, p. 1312, Mar. 2019.

[31] D. Roetenberg, H. J. Luinge, C. T. M. Baten, and P. H. Veltink, "Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 13, no. 3, pp. 395–405, Sep. 2005.

[32] Z. Pentek, T. Hiller, and A. Czmerk, "Algorithmic enhancement of automotive MEMS gyroscopes with consumer-type redundancy," *IEEE Sensors J.*, vol. 21, no. 2, pp. 2092–2103, Jan. 2021.

[33] B. Nagy, J. Botzheim, and P. Korondi, "Magnetic angular rate and gravity sensor based supervised learning for positioning tasks," *Sensors*, vol. 19, no. 24, p. 5364, Dec. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/24/5364

[34] S. Yue, L. Cong, H. Qin, B. Li, and J. Yao, "A robust fusion methodology for MEMS-based land vehicle navigation in GNSS-challenged environments," *IEEE Access*, vol. 8, pp. 44087–44099, 2020.

[35] W. J. Park, J. W. Song, C. H. Kang, J. H. Lee, M. H. Seo, S. Y. Park, J. Y. Yeo, and C. G. Park, "MEMS 3D DR/GPS integrated system for land vehicle application robust to GPS outages," *IEEE Access*, vol. 7, pp. 73336–73348, 2019.

[36] T. Duriez, S. L. Brunton, and B. R. Noack, *Machine Learning Control-Taming Nonlinear Dynamics and Turbulence*. Cham, Switzerland: Springer, 2017.

[37] J. Kusaka, T. Obo, J. Botzheim, and N. Kubota, "Joint angle estimation system for rehabilitation evaluation support," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2014, pp. 1456–1462.

[38] D. Zhou, Y. Fang, J. Botzheim, N. Kubota, and H. Liu, "Bacterial memetic algorithm based feature selection for surface EMG based hand motion recognition in long-term use," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–7.

[39] J. Botzheim, C. Cabrita, L. T. Koczy, and A. E. Ruano, "Genetic and bacterial programming for B-spline neural networks design," *J. Adv. Comput. Intell. Inform.*, vol. 11, no. 2, pp. 220–231, 2007.

[40] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[41] N. E. Nawa and T. Furuhashi, "Fuzzy system parameters discovery by bacterial evolutionary algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 5, pp. 608–616, Oct. 1999.

[42] K. Balazs, J. Botzheim, and L. T. Koczy, "Hierarchical fuzzy system modeling by genetic and bacterial programming approaches," in *Proc. Int. Conf. Fuzzy Syst.*, Jul. 2010, pp. 1–6.

[43] A. A. Saputra, J. Botzheim, and N. Kubota, "Evolving a sensory–motor interconnection structure for adaptive biped robot locomotion," *IEEE Trans. Cogn. Developmental Syst.*, vol. 11, no. 2, pp. 244–256, Jun. 2019.

ZSIGMOND PÉNTEK received the M.Sc. degree in mechatronics of precision systems from the University of Technology and Economics, Budapest, Hungary, in 2014. He is currently pursuing the Ph.D. degree with Bosch in a cooperation with the Budapest University of Technology and Economics, Budapest, Hungary. Since 2014, he works with Robert Bosch Ltd., Budapest, where he is involved in MEMS gyroscope development and characterization. His research interests include autonomous system calibration, processing of sensor signals for control of robotic manipulators with a focus on MEMS gyroscope sensor performance.

JÁNOS BOTZHEIM (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the Budapest University of Technology and Economics, in 2001 and 2008, respectively. He is an Associate Professor with the Department of Mechatronics, Optics and Mechanical Engineering Informatics, Budapest University of Technology and Economics, Hungary. He is a member of several scientific societies such as John von Neumann Computer Science Society, and Hungarian Fuzzy Association. His research interests are computational intelligence and cognitive robotics.

ANDRÁS CZMERK received the Ph.D. degree in modeling and control of pneumatic systems from the Budapest University of Technology and Economics, Budapest, Hungary, in 2015. Since 2008, he works with the Budapest University of Technology and Economics, Budapest, where he teaches mechatronics, measurement techniques and automation in mechanical engineering. His research interests include control and modeling of nonlinear pneumatic systems, and system automation in industry 4.0 in a cooperation with Robert Bosch Ltd.