*electronics*

MDPI

*Article*

# Dynamic Modeling and Simulation of a Four-Wheel Skid-Steer Mobile Robot Using Linear Graphs

**Eric McCormick [1], Haoxiang Lang [1,*] and Clarence W. de Silva [2]**

[1] GRASP Lab, Departmet of Automotive and Mechatronics Engineering, Ontario Tech University, Oshawa, ON L1G 0C5, Canada
[2] Mechanical Engineering Department, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[*] Correspondence: haoxiang.lang@ontariotechu.ca

**Abstract:** This paper presents the application of the concepts and approaches of linear graph (LG) theory in the modeling and simulation of a four-wheel skid-steer mobile robotic system. An LG representation of the system is proposed, and the accompanying state-space model of the dynamics of a mobile robot system is evaluated using the associated LGtheory MATLAB toolbox, which was developed in our lab. A genetic algorithm (GA)-based parameter estimation method is employed to determine the system parameters, which leads to a very accurate simulation of the model. The developed model is then evaluated and validated by comparing the simulated LG model trajectory with the trajectory of an ROS Gazebo-simulated robot and experimental data obtained from the physical robotic system. The obtained results demonstrate that the proposed LG model, combined with the GA parameter estimation process, produces a highly accurate method of modeling and simulating a mobile robotic system.

**Keywords:** dynamic system modeling; genetic algorithms; linear graph; mobile robots

check for updates

**Citation:** McCormick, E.; Lang, H.; de Silva, C.W. Dynamic Modeling and Simulation of a Four-Wheel Skid-Steer Mobile Robot Using Linear Graphs. *Electronics* **2022**, *11*, 2453. https://doi.org/10.3390/electronics11152453

Academic Editor: Cecilio Angulo

Received: 2 July 2022
Accepted: 4 August 2022
Published: 6 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## 1. Introduction

Engineering systems are becoming increasingly more advanced with the integration of multiple physical domains, such as mechanical, electrical, fluid and thermal domains, into systems in an integrated and unified manner, which may have traditionally only been considered interconnected independent subsystems, each of which comprised just a single physical or energy domain. The resulting field of multi-disciplinary engineering, known as mechatronic engineering, has brought with it a significant acceleration in technological advancement. While this advancement has come with parallel advancements in computer-automated robotic, vehicular, machine tool, and electronic systems as well as more sophisticated control, this has also resulted in an increase in the complexity associated with modeling, simulating, designing, and controlling such multi-domain engineering systems. In recent decades, many methods of system modeling have been introduced to address this complexity issue, but most of them lack the necessary integrated and unified focus of mechatronics, which can be facilitated by graph-based modeling methods. In particular, in the 1960s, Henry M. Paynter introduced two separate graph-based methods of dynamic system modeling: linear graphs (LGs) and bond graphs (BGs) [1].

While BG modeling has seemingly surpassed the popularity of LG modeling, due in part to the development of commercial software tools such as 20sim, which facilitate the process of evaluating complex BG models [2–5], the LG approach provides some additional benefits beyond those associated with BG theory.

The key benefit that the LG approach provides over the BG approach is the intuitive nature in which LG models can be constructed and recognized so that the model structure directly corresponds to the actual physical structure of the system. Figure 1 shows an example of modeling a mechanical system using LG and BG. Notably, the electrical and mechanical systems will have analogous structures in their LG models. For many energy domains, such as electrical and fluid, the conversion of their schematic diagrams to LG models can often result in a nearly identical topological layout.

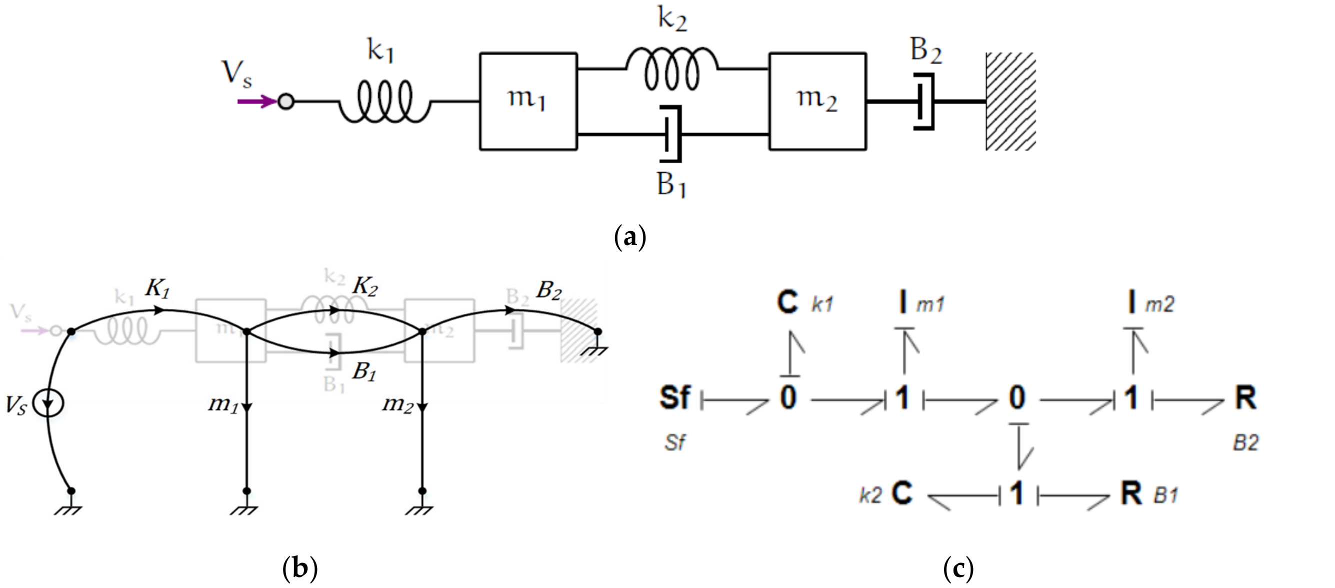


**Figure 1.** (**a**) Schematic model of a mass-spring-damper system. (**b**) Equivalent LG model. (**c**) Equivalent BG model.

Additionally, the use of simple node and loop network equations, such as those of Kirchhoff's current and voltage laws, allow for the analogous application of electrical network-like algorithms across systems consisting of many energy domains, leading to analogous loop equations and analogous node equations. Similarly, the network representation provided by the LG approach provides a more intuitive method of determining the power flow of the system, thus allowing for an easier method of determining the dependent and independent variables of the system. Furthermore, LG modules can be introduced to represent such physical devices as amplifiers by means of modulated source elements, converters from the energy domain to the information domain (e.g., process to sensor), and converters from the information domain to the energy domain (e.g., control to actuator) [6].

LG theory, evolved from Leonhard Euler's graph theory, was first applied in engineering for the analysis of large electrical networks before being expanded to applications spanning multiple energy domains [7]. Before LG theory, the analysis of multi-domain systems was neither integrated nor unified, meaning that different physical domains were modeled separately without considering their dynamic interactions, and different (non-analogous) techniques were used for modeling each domain. While initial developments in LG theory focused on the primary energy domains (electrical, mechanical, fluid, and thermal), developments in additional domains, such as thermohydraulic [8,9], electrochemical [10], and multibody [11,12], started to emerge. While the recent work in the field of LGs is rather limited when compared with BGs, some research is being conducted on the application of LG theory for automatic design evolution [13].

To demonstrate the LG approach and its versatility and robustness, this paper develops an LG model representation of the dynamics of a four-wheel skid-steer mobile robot and verifies the accuracy by comparing the physical system and existing model provided in a popular robotics simulator (Gazebo). A MATLAB toolbox, named the LG theory MATLAB toolbox, has been developed in our laboratory to address the lack of available LG-based software tools [14]. This toolbox will be utilized to automate the process of formulating the state-space model of the dynamic system. This model will be further enhanced with a genetic algorithm (GA)-based parameter estimation procedure in order to calibrate the unknown parameter values of the model. Evaluation and validation of the proposed model will be performed via comparisons of the trajectory response of the simulated LG model against the data collected from a robot operating system (ROS)-based Gazebo simulation and from experimental data of real-world driving scenarios using a Clearpath Husky mobile robotic system.

The research problem of the paper is to address mechatronic engineering concerns on the development of integrated (or concurrent) design approaches with synergistic consideration of multidisciplinary (or multi-domain or multi-physics) systems using "unified" approaches (involving analogous methodologies across physical domains) in a systematic manner to arrive at a "unique" solution (which implicitly means the use of an optimization scheme). The objective of the proposed approach in this paper will lead to improvements in efficiency, reliability, component compatibility, ease of system design, expansion, and enhancement while reducing the system complexity.

## 2. Linear Graph Modeling

The LG approach is a method of modeling and evaluating complex dynamic systems through the use of a simplistic graphical representation in order to derive their state-space models. This is a systematic, unique, integrated, and unified approach which uses a well-established set of steps (systematic) leading to a single model (unique) by considering all physical domains and their interactions simultaneously (integrated) and using analogous methods to model the different domains (unified). Essentially, it provides a robust modeling method which produces a single unique model for a specific system through the analogous application of methodologies across multiple energy domains. This means that a multi-domain mechatronic system is evaluated while not using a single integrated model as a series of separate models (which is the "sequential" approach) and while applying similar network equations and algorithms for each of the system's energy domains [6].

The procedure introduced in [15] can be applied to LG representation of the physical model to derive a standard form of state-space model of the modeled multi-domain engineering system as shown in Equation (1):

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} \\ \boldsymbol{y} &= C\boldsymbol{x} + D\boldsymbol{u} \end{aligned} \tag{1}$$

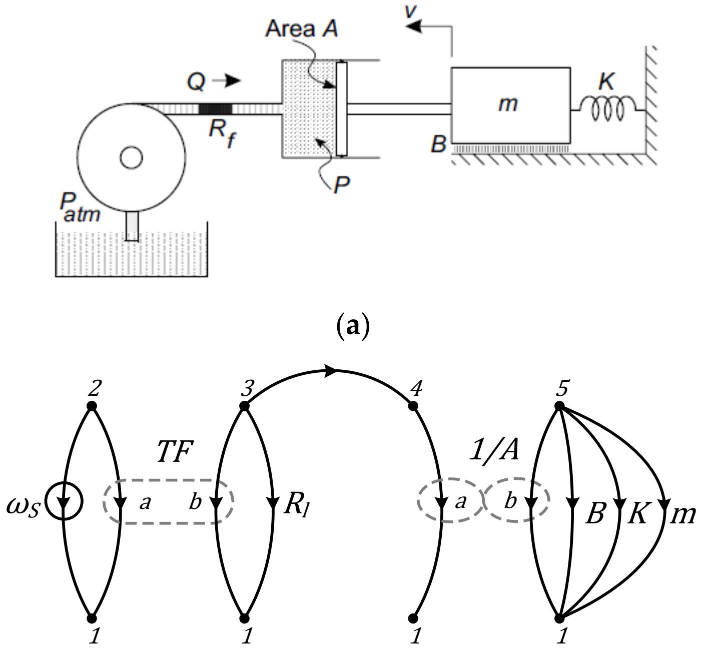
The conversion between the graphical LG model and the mathematical state-space model was achieved by using the developed LGtheory Matlab Toolbox in our lab [14].

Figure 2 shows an example of modeling a hydro-mechanical system consisting of an electric motor to power a positive-displacement pump and piston, which actuate a mass element attached to a spring, and to stay grounded in the mechanical translation domain.

The following equations represent the state-space model produced by the toolbox, with outputs specified as the pressure of the fluid between the pump and the piston and the velocity of the mass element:

$$\begin{gathered} \begin{bmatrix} \dot{v_m} \\ \dot{F_k} \end{bmatrix} = \begin{bmatrix} \frac{A^2\left(R_f+R_l-R_fR_lTF\right)}{m(R_lTF-1)} - \frac{B}{m} & -\frac{1}{m} \\ K & 0 \end{bmatrix} \begin{bmatrix} v_m \\ F_k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} [\omega_s] \\ \begin{bmatrix} P_{R_f} \\ v_m \end{bmatrix} = \begin{bmatrix} -AR_f & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_m \\ F_k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} [\omega_s] \end{gathered} \tag{2}$$

With the successful development and validation of an automated tool for evaluating LG models, this approach can now be employed to facilitate more advanced applications, such as the automated evolutionary design of engineering systems, the modeling of larger more complex multi-domain dynamic systems, and the monitoring and design optimization of complex multi-domain mechatronic systems.



(**a**)

(**b**)

**Figure 2.** (**a**) System model and (**b**) LG model.

## 3. Development of the Linear Graph Model for Mobile Robots

The physical four-wheel skid-skip mobile robot (ClearPath Husky) and the diagram of its subsystems are shown in Figure 3. The LG model of the robot consists of various subsystems that encompass multiple physical domains and functions of the robotic system. The complete model includes the electrical subsystem, consisting of DC motors powered by a voltage source (battery), the drivetrain subsystems, consisting of the front and rear axles and wheels for both of the independent left and right side powertrains, and the translational and rotational subsystems, representing the linear and rotational movements of the entire mobile robot.

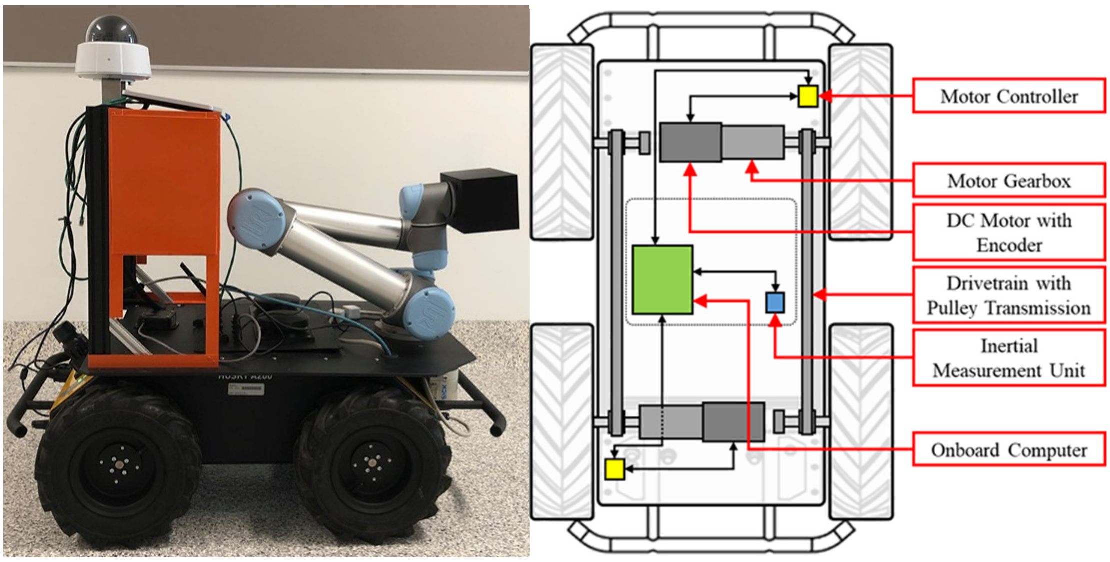


**Figure 3.** ClearPath Husky mobile robotic system and the diagram of its subsystems.

### *3.1. Electrical Subsystem*

The Husky's electrical subsystem consists of two 24-V brushed DC motors, each with a 78.71:1 gearbox reduction attachment which transmits the output torque of the motors to the primary axles of the two independent drivetrains. Each DC motor drive module is modeled as a series LG circuit, consisting of a controlled DC voltage source ($V_{S1}$), a D-type resistive element ($R_1$), a T-type inductive element ($L_1$), and a two-port transformer element representing the combined output of the motor torque constant and the gearbox reduction ($T_{ML}$).

The equations produced by the LGtheory toolbox for the electrical subsystem based on the constitutive equations and the normal tree of the LG model are as follows:

The continuity equations for each passive branch are:

$$i_{R_1} = i_{T_{ML}} = i_{L_1} \tag{3}$$

The compatibility equation for each loop formed by the temporary inclusion of each passive link is:

$$V_{L_1} = V_{s1} - V_{T_{ML}} - V_{R_1} \tag{4}$$

### *3.2. Drivetrain and Wheel Subsystem*

The drivetrain subsystem of Husky consists of two independent drivetrains, each powered by its own DC motor. Each of these independent drivetrains consists of a primary axle, to which the DC motor is directly connected, and a secondary axle, which receives power from the primary axle via a 1:1 belt drive system. Each of these independent drivetrains powers both of the wheels on their respective sides of the vehicle. Figure 4 shows a diagram of the system configuration of Husky, with the drivetrain components and sensors used for data collection.

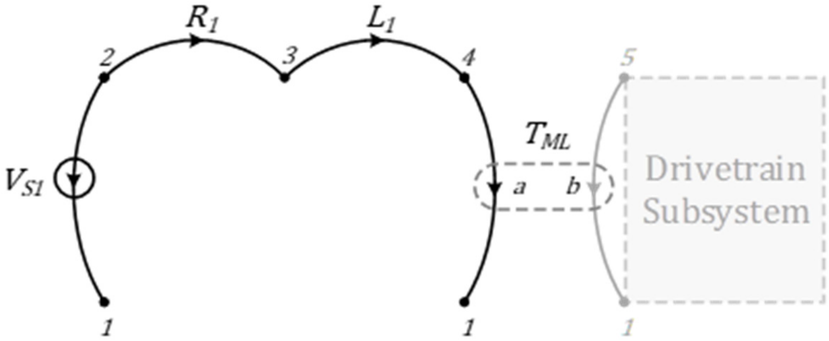


**Figure 4.** LG model and the normal tree of the Husky robot electrical subsystem.

The present paper presents two methods for modeling the drivetrain subsystem. The first method, termed the expanded model, considers the dynamics of all the wheels on the vehicle separately and models the compliance and slippage of the belt drive system. The second method, termed the simplified model, assumes the belt drive system is not flexible and models the inertias of the two axles of each drivetrain as a single element.

For the expanded model (considering only the drivetrain on the left-hand side of the vehicle here, as the other half of the drivetrain is modeled in the same manner), the second branch of the two-port element (coming from the corresponding motor of the electrical subsystem) splits at its upper node into four paths. The first two of these paths, which are set in parallel with the second port of the motor transformer, consist of elements that represent the parameters of the primary axle of the drivetrain. The first path contains an A-type element ($J_{RL}$), and the second path consists of a D-type element ($B_{RL}$) in series with two transformer elements ($TF_1$ and $TF_2$). The first path represents the combined inertial load of the wheel and shaft of the primary axle, whereas the second path collectively represents all of the energy that is lost or transferred out of the drivetrain system. The

D-type element represents the energy dissipation from the system due to friction in the shaft bearings and also the friction of the wheels slipping on the driving surface, and the transformer elements represent the transmission of torque from the drivetrain subsystem to the translational and rotational dynamics subsystems of the Husky vehicle. The third and fourth paths consist of a D-type ($B_{BeltL}$) element and a T-type ($K_{BeltL}$) element in parallel, representing the compliance and slip of the belt in the pulley transmission system, respectively. These paths then split at their shared second node into two more paths, representing to the inertial load ($J_{FL}$) and the energy dissipation and transmission ($B_{FL}$, $TF_3$, and $TF_4$) of the secondary axle, similar to the primary axle.

Figure 5 illustrates the mentioned benefits of LG modeling over other modeling approaches. The structure of the LG model has a closer resemblance to the physical system than any other form of dynamic system modeling, such as a BG model. In particular, the LG model shows how the elements represent the dynamics of each axle and overlay their respective wheels, while the parallel D-type and T-type elements closely resemble the belt that connects the two shafts.

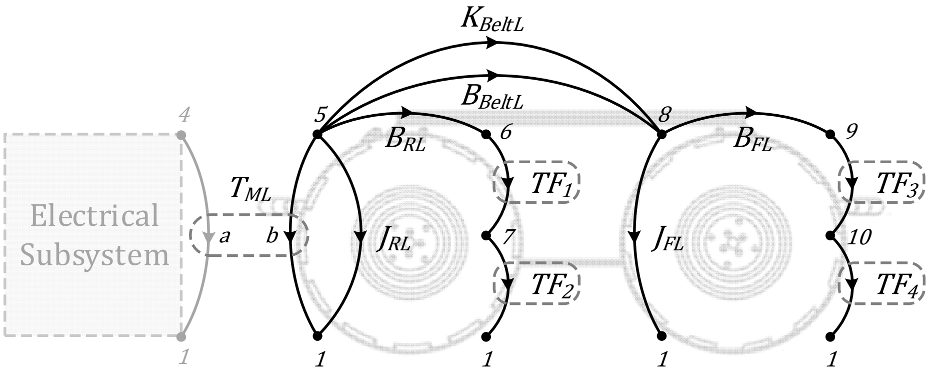


**Figure 5.** LG model of the left-side drivetrain subsystem overlaid on the profile of the Husky robot.

For the simplified model, it is assumed that the belt is not flexible, thus eliminating the compliance between the primary and secondary axles of the drivetrain. With this change, the model can be further simplified by combining the inertia of the primary and secondary axels into a single element ($J_L$) as shown in Figure 6.

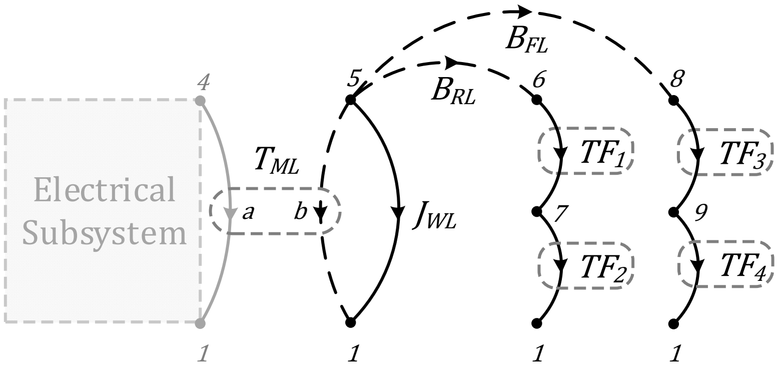


**Figure 6.** Simplified LG model and the normal tree of the left-side drivetrain subsystem.

The equations produced by the LGtheory toolbox for the simplified drivetrain subsystem based on the constitutive equations and normal tree of the LG model are as follows.

The continuity equations for each passive branch are

$$\begin{aligned} \tau_{J_{LW}} &= -\tau_{B_{FL}} - \tau_{B_{RL}} - \tau_{T_{ML}} \\ \tau_{TF_1} &= \tau_{TF_2} = \tau_{B_{RL}} \\ \tau_{TF_3} &= \tau_{TF_4} = \tau_{B_{FL}} \end{aligned} \tag{5}$$

The compatibility equations for each loop formed by the temporary inclusion of each passive link are

$$\begin{aligned} \omega_{T_{ML}} &= \omega_{J_{LW}} \\ \omega_{B_{RL}} &= \omega_{J_L} - \omega_{TF_1} - \omega_{TF_2} \\ \omega_{B_{FL}} &= \omega_{J_L} - \omega_{TF_3} - \omega_{TF_4} \end{aligned} \tag{6}$$

Drivetrain Transformer Equations

The four transformer elements of the half drivetrain model shown in Figure 7 represent the conversion of the wheel torque into the traction force that propels the vehicle linearly ($TF_{odd}$) and the conversion of the torque and traction force of the wheel into the moment that rotates the vehicle ($TF_{even}$). The equations that define these power conversions are determined next.

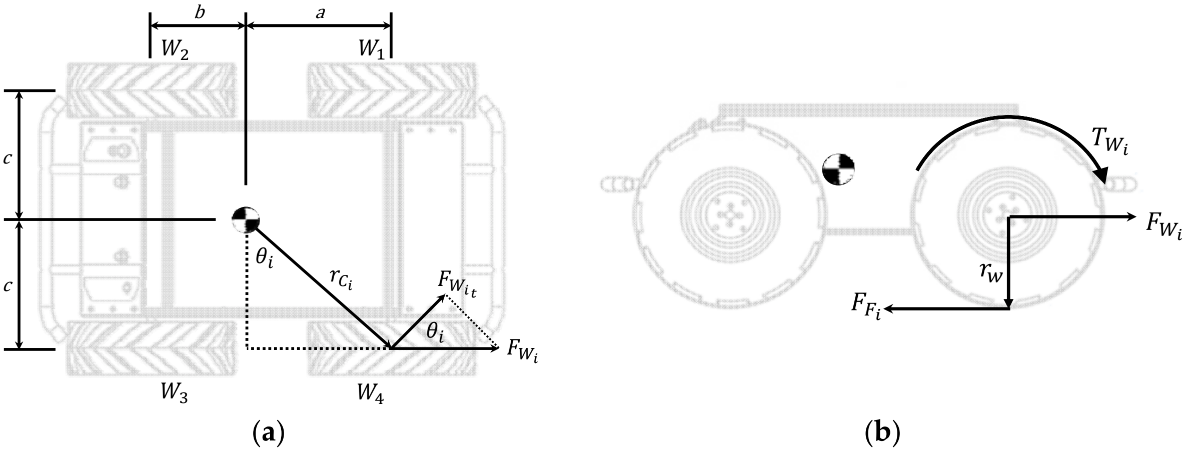


**Figure 7.** Diagrams of (**a**) vehicle rotational moments and (**b**) wheel tractive forces produced by the torque of the wheels.

For $TF_{odd}$, the rotation of the wheels of the vehicle due to the power provided by the DC motors creates friction between the tread of the tire and the driving surface ($F_{F_i}$). This traction between the two surfaces creates a counter force on the vehicle ($F_{W_i}$), which propels the vehicle forward.

Referring to the constitutive equations for a transformer, $f_1$ is the generalized through-variable of the first port, representing the wheel torque, and $f_2$ is the generalized through-variable of the second port, representing the traction force of the wheels. In order to convert the wheel torque into a force, $f_1$ must be divided by the radius of the wheel ($r_W$):

$$TF_{odd} = \frac{1}{r_W} \tag{7}$$

Similarly, for $TF_{even}$, the traction force that propels the vehicle which is generated by the torque of the DC motors also produces the moment that rotates the vehicle.

Referring to the same constitutive equations for a transformer as before, $f_1$ is the generalized through-variable of the first port, representing the wheel torque, and $f_2$ is the generalized through-variable of the second port, representing the moment applied on the vehicle. In this transformer, two conversions occur: the first is the conversion of the wheel torque into traction force, and the second is the conversion of the traction force into the rotational moment of the vehicle. Again, to create the traction force, the torque of the wheel must be divided by the radius of the wheel. To convert this traction force into rotational torque, the distances from the center of mass of the vehicle to the contact point of each wheel and the driving surface ($r_{c_i}$) must be found. The rotational torque of the vehicle produced by each wheel can then be determined by multiplying these distances by the tangential component of the force of each wheel (where Equation (8) is "+" for right side wheels and "−" for left):

$$TF_{even} = \pm cos\left(\theta_{W_i}\right)\cdot r_{C_i}\cdot \frac{1}{r_W} \tag{8}$$

Here, $TF_2$ corresponds to the rear left wheel ($W_2$), $TF_4$ the front left wheel ($W_1$), $TF_6$ the front right wheel ($W_4$), and $TF_8$ the rear right wheel ($W_3$), as shown in Figure 7. Therefore, for $TF_2$ and $TF_8$, the following equations apply:

$$\theta_{W_2} = \theta_{W_3} = tan^{-1}\left(\frac{b}{c}\right) \tag{9}$$

$$r_{C_2} = r_{C_3} = \sqrt{(b)^2 + (c)^2} \tag{10}$$

Additionally, for $TF_4$ and $TF_6$, the following equations apply:

$$\theta_{W_1} = \theta_{W_4} = tan^{-1}\left(\frac{a}{c}\right) \tag{11}$$

$$r_{C_1} = r_{C_4} = \sqrt{(a)^2 + (c)^2} \tag{12}$$

### 3.3. Mobile Robot Translational Dynamic Subsystem

The translational dynamics of the Husky robot are determined by the summation of the traction forces produced by the wheels of the vehicle as shown in Figure 8. This is represented in LG form by placing the second ports of the odd-numbered transformer elements in parallel with an A-type element ($M_{Husky}$) representing the mass of the Husky vehicle. This configuration means that the forces produced by each tire will be summed in order to induce an acceleration in the vehicle when the wheels produce a non-zero resultant force and allow the vehicle to remain stationary when the tire forces are balanced (i.e., the vehicle is stopped or turning on the spot).

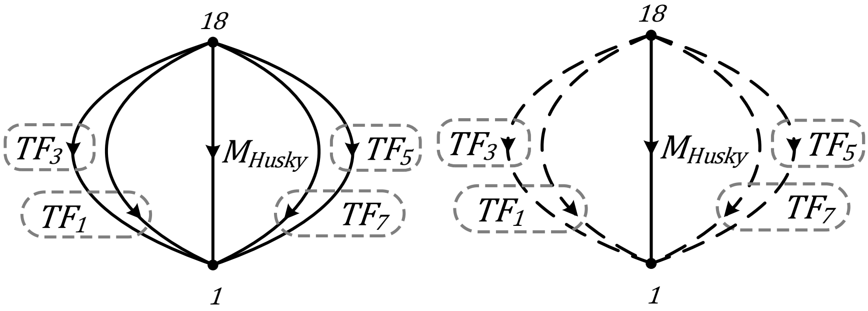


**Figure 8.** LG model and the normal tree of the Husky robot translational dynamic subsystem.

The equations generated by the LGtheory toolbox for the translational dynamic subsystem of the mobile robot based on the constitutive equations and the normal tree of the LG model are as follows.

The continuity equation for each passive branch is

$$F_{M_{Husky}} = -F_{TF_1} - F_{TF_3} - F_{TF_5} - F_{TF_7} \tag{13}$$

The compatibility equations for each loop formed by the temporary inclusion of each passive link are

$$v_{TF_1} = v_{TF_3} = v_{TF_5} = v_{TF_7} = v_{M_{Husky}} \tag{14}$$

### 3.4. Mobile Robot Rotational Dynamics Subsystem

The rotational dynamics of the Husky robot are determined by the summation of the rotational moments produced by the wheels of the vehicle as shown in Figure 9. This is represented in LG form by placing the second ports of the even-numbered transformer elements in parallel with a D-type element ($B_{Husky}$) and an A-type element ($J_{Husky}$), representing the resistance to rotational movement and the inertia of the Husky vehicle, respectively. This configuration means that the torques produced by each tire will be summed in order to induce a rotation on the vehicle when the wheels produce a non-zero resultant moment and allow the vehicle to remain stationary when the rotational moment is balanced (i.e., the vehicle is not turning).

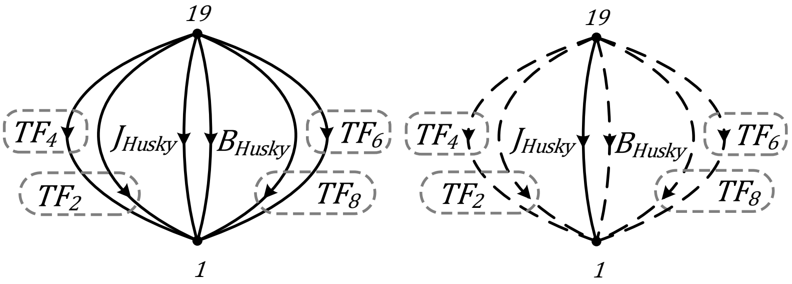


**Figure 9.** LG model and the normal tree of the Husky robot's rotational dynamic subsystem.

The equations produced by the LGtheory toolbox for the mobile robot rotational dynamics subsystem based on the constitutive equations and the normal tree of the LG model are given next.

The continuity equation for each passive branch is

$$\tau_{J_{Husky}} = -\tau_{TF_2} - \tau_{TF_4} - \tau_{TF_6} - \tau_{TF_8} - \tau_{B_{Husky}} \tag{15}$$

The compatibility equations for each loop formed by the temporary inclusion of each passive link are

$$\begin{gathered} \omega_{TF_2} = \omega_{TF_4} = \omega_{TF_6} = \omega_{TF_8} = \omega_{J_{Husky}} \\ \omega_{B_{Husky}} = \omega_{J_{Husky}} \end{gathered} \tag{16}$$

### 3.5. Complete Linear Graph Model of the Mobile Robot

The complete LG model of the Husky robot is produced by combining the LG models of the subsystems presented in the previous section. The results for the LG model presented in Figure 10 encompass the dynamics of the entire Husky system.

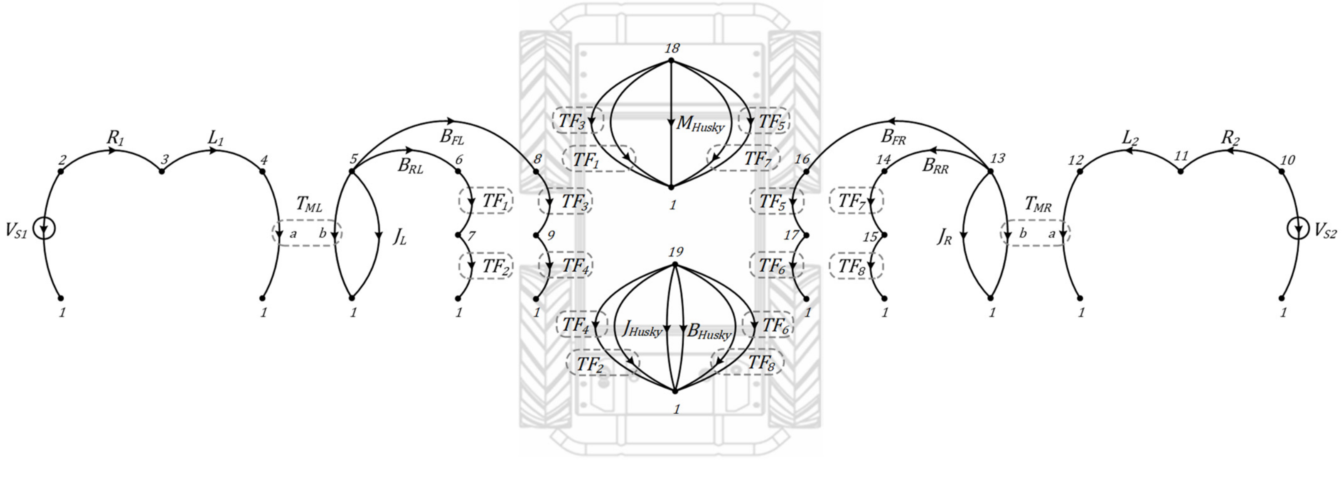


**Figure 10.** Complete LG model of the ClearPath Husky robot with the simplified drivetrain subsystems.

By evaluating the sets of equations produced by the various subsystem models along with the corresponding equations for the right-side powertrain and the constitutive equations for each element, the LGtheory toolbox generates the following state-space model for the Husky system. Note that the "Husky" subscript has been replaced with "*H*":

$$\frac{dx}{dt} = A\begin{bmatrix} \omega_{J_{LW}} \\ \omega_{J_{RW}} \\ v_{M_H} \\ \omega_{J_H} \\ i_{L_1} \\ i_{L_2} \end{bmatrix} + B\begin{bmatrix} V_{s1} \\ V_{s2} \end{bmatrix} \tag{17}$$

$$y = C\begin{bmatrix} \omega_{J_{LW}} \\ \omega_{J_{RW}} \\ v_{M_H} \\ \omega_{J_H} \\ i_{L_1} \\ i_{L_2} \end{bmatrix} + D\begin{bmatrix} V_{s1} \\ V_{s2} \end{bmatrix} \tag{18}$$

$$A = \begin{bmatrix} \frac{-B_{FL}-B_{RL}}{J_L} & 0 & \frac{B_{FL}TF_3+B_{RL}TF_1}{J_L} & \frac{B_{FL}TF_4+B_{RL}TF_2}{J_L} & \frac{T_{ML}}{J_L} & 0 \\ 0 & \frac{-B_{FR}-B_{RR}}{J_R} & \frac{B_{FR}TF_5+B_{RR}TF_7}{J_R} & \frac{B_{FR}TF_6+B_{RR}TF_8}{J_R} & 0 & \frac{T_{MR}}{J_R} \\ \frac{B_{FL}TF_3+B_{RL}TF_1}{M_H} & \frac{B_{FR}TF_5+B_{RR}TF_7}{M_H} & \frac{-B_{RL}TF_1^2-B_{FL}TF_3^2-B_{FR}TF_5^2-B_{RR}TF_7^2}{M_H} & \frac{-B_{FL}TF_3TF_4-B_{FR}TF_5TF_6-B_{RL}TF_1TF_2-B_{RR}TF_7TF_8}{M_H} & 0 & 0 \\ \frac{B_{FL}TF_4+B_{RL}TF_2}{J_H} & \frac{B_{FR}TF_6+B_{RR}TF_8}{J_H} & \frac{-B_{FL}TF_3TF_4-B_{FR}TF_5TF_6-B_{RL}TF_1TF_2-B_{RR}TF_7TF_8}{J_H} & \frac{-B_{RL}TF_2^2-B_{FL}TF_4^2-B_{FR}TF_6^2-B_{RR}TF_8^2-B_H}{J_H} & 0 & 0 \\ -\frac{T_{ML}}{L_1} & 0 & 0 & 0 & -\frac{R_1}{L_1} & 0 \\ 0 & -\frac{T_{MR}}{L_2} & 0 & 0 & 0 & -\frac{R_2}{L_2} \end{bmatrix} \tag{19}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{L_1} & 0 \\ 0 & \frac{1}{L_2} \end{bmatrix} \tag{20}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{21}$$

$$D = [0]_{4\times 2} \tag{22}$$

The outputs of the state-space model are chosen to be the state variables representing the rotational velocities of the left and the right wheels, as well as the linear and rotational velocities of the mobile robot itself. The selection of these outputs allows for a comparison of these states with the corresponding Gazebo-simulated and experimental state values of the robot for the purposes of model validation and parameter estimation.

### 3.6. Parameter Estimation Using the Genetic Algorithms

GAs are a form of multi-point, population-based methods for simultaneously exploring multiple solutions to an optimization problem. Based on the same concepts as natural evolution, GAs reproduce and evolve members of a population, referred to as solutions, over many generations in order to obtain an optimized solution to a problem. Throughout this evolutionary process, new solutions inherit the beneficial characteristics from the successful solutions of past generations while also introducing new characteristics that may provide advantages over other solutions. Those solutions that are successful are more likely to reproduce and thus pass on their beneficial characteristics, while those that are less successful face the possibility of being purged from the population. This process tends to lead to a stronger population of solutions over many generations. GAs are stochastic in nature, and therefore they do not necessarily guarantee finding the most optimal solution to a problem. Rather, this method is useful for more complex optimization problems that are difficult or infeasible to solve through mathematical means [16].

The present paper utilizes the GA capabilities of the Global Optimization Toolbox of MATLAB. The objective function of the parameter estimation GA is based on the sum of the absolute tracking errors in the x and y directions between the LG-based mobile robot simulation and the data obtained from Gazebo and the physical experiments using the Husky robot:

$$obj = \sum |x_{data}(t) - x_{LG}(t)| + \sum |y_{data}(t) - y_{LG}(t)| \tag{23}$$

While most of the system parameters of the Husky robot can be determined from the manufacturer documentation or from the component datasheets, some system parameters that are based on specific environmental conditions must be calibrated for. These parameters, which will be calibrated using a GA, are the unknown damping constant values given in Table 1, representing the losses of the drivetrain systems due to slip and friction ($B_{LW},\ B_{RW}$) and the resistance to the rotational movement between the wheels of the Husky vehicle and the driving surface ($B_{Husky}$). Additionally, the GA will calibrate the coefficient value ($c$) of a simple multivariable function, which is used to estimate the motor voltage signals from the recorded command velocity signals sent to the robot. Since real-time measurements of the motor voltages cannot be obtained at a sufficiently fast rate from the ROS for the physical experiments, the following functions have to be utilized in order to estimate the voltage inputs for the state-space model:

$$V_{s1} = 24 \cdot (c \cdot v_t + 0.541 \cdot c \cdot v_r) \tag{24}$$

$$V_{s2} = 24 \cdot (c \cdot v_t - 0.541 \cdot c \cdot v_r) \tag{25}$$

**Table 1.** State-space parameters, including unknown values.

| Description | Parameter | Value | Units |
|---|---|---|---|
| Voltage Inputs | $V_{s1},\ V_{s2}$ | ±24 | V |
| Internal Motor Resistance | $R_1, R_2$ | 0.46 | Ω |
| Internal Motor Inductance | $L_1, L_2$ | 0.22 | mH |
| Motor Torque Constant | $k_t$ | 0.044488 | N·m/A |
| Gear Ratio | $GR$ | 78.71:1 | Gear Ratio |
| Motor Transformer Ratio | $T_{ML}, T_{MR}$ | $k_t \times GR$ | N·m/A |
| Drivetrain Inertia | $J_{LW}, J_{RW}$ | 0.08 | kg·m$^2$ |
| Drivetrain Damping | $B_{RL,FL,FR,RR}$ | Unknown | rad/(N·m·s) |
| Power Conversion | $TF_{odd}$ | Equation (7) | |
| Transformer Ratios | $TF_{even}$ | Equation (8) | |
| Husky Mass | $M_{Husky}$ | 48.39 | kg |
| Husky Rotational Damping | $B_{Husky}$ | Unknown | rad/(N·m·s) |
| Husky Inertia | $J_{Husky}$ | 3.0556 | kg·m$^2$ |

Here, $v_t$ and $v_r$ are the command values for the translational and rotational velocities, respectively. It was found through experimentation with the command signals to the motor that a rotational velocity command which was numerically equivalent to a translational velocity command would result in 54.1% rotational speed of the Husky wheels, and hence the inclusion of the 0.541 values in Equations (24) and (25).

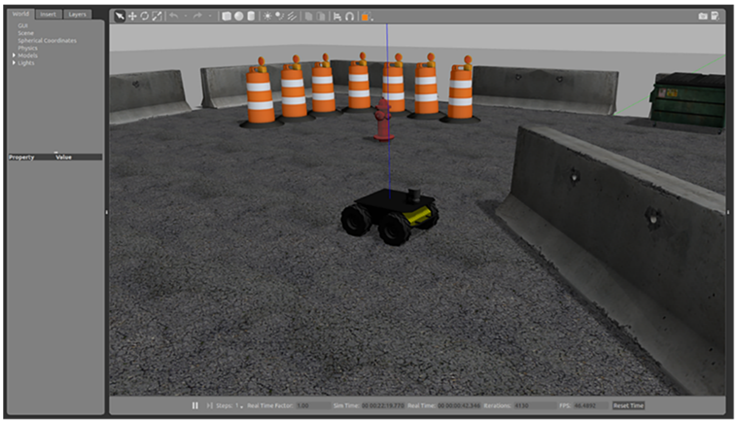
Table 1 presents the known and unknown parameter values for the mobile robot state-space model.

## 4. Results and Discussion

### *4.1. ROS Gazebo Simulation Environment*

Gazebo is a physics-based 3D environment for simulating the rigid body dynamics of robotic systems as shown in Figure 11. Along with dynamic simulations, Gazebo can be used for simulating sensor readings, evaluating and training artificial intelligence

(AI)-based control systems, and much more. Integration of Gazebo with an ROS allows for exploiting the data transfer and communication capabilities which the ROS facilitates. These capabilities facilitate recording of the command signals and sensor data of the robot using the following ROS topics: /tf (the transformation frames of the robot used for linear speed and trajectory), /imu_um7/data (the IMU data readings for rotational speed of the Husky) /joint_states (the encoder readings of each wheel), and /husky_velocity_controller/cmd_vel (the command signals sent to the robot from the gamepad controller). The ROS then provides this recorded data to be filtered for the command data and replayed as the input to a Gazebo simulation in order to seek replicating the results.



Steps: 1 Real Time Factor: 1.00 Sim Time: 00 00:22:19.770 Real Time: 00 00:00:42.346 Iterations: 4130 FPS: 46.4892

**Figure 11.** Example of a Gazebo simulation environment.

### *4.2. GA Estimation and Comparison of Results*

Initial calibration of the Husky LG model was conducted using data collected from the physical robot while executing a circular maneuver. This maneuver involved an initial straight trajectory before entering a circular trajectory. Once a full revolution was complete, the Husky exited the circle again in a straight path.

Figure 12 shows the performance of the GA-based parameter estimation procedure. The procedure used to calibrate the model was run with a population of 100 solutions for a maximum of 100 generations and with a crossover fraction of 0.5. Table 2 gives the parameters that are estimated, the upper and lower bounds of their searches, and the results obtained which produced the optimal simulation results.

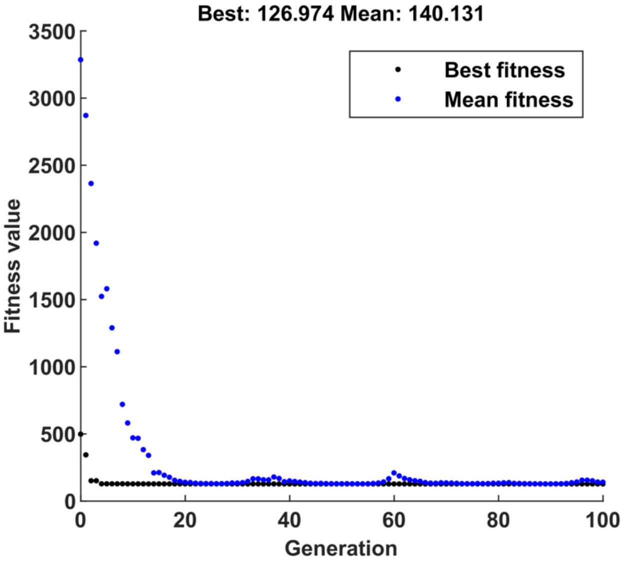


**Figure 12.** GA evolution of parameter estimation.

**Table 2.** Variables adapted by the GA and the resulting values.

| Variable | $B_{FL}$,$B_{RL}$,$B_{RR}$,$B_{FR}$ | $B_{Husky}$ | $c$ |
|---|---|---|---|
| Upper Bounds | 1 | 1 | 0.75 |
| Lower Bounds | 100 | 100 | 1.00 |
| Results | 1.3016 | 12.8650 | 0.8961 |

Figure 13 shows the response of the state-space outputs specified for the LG model against the corresponding measured data from the Husky vehicle for the duration of the circular maneuver. As can be seen, there was a strong conformance between the response of the simulated output states and the Husky sensor readings.

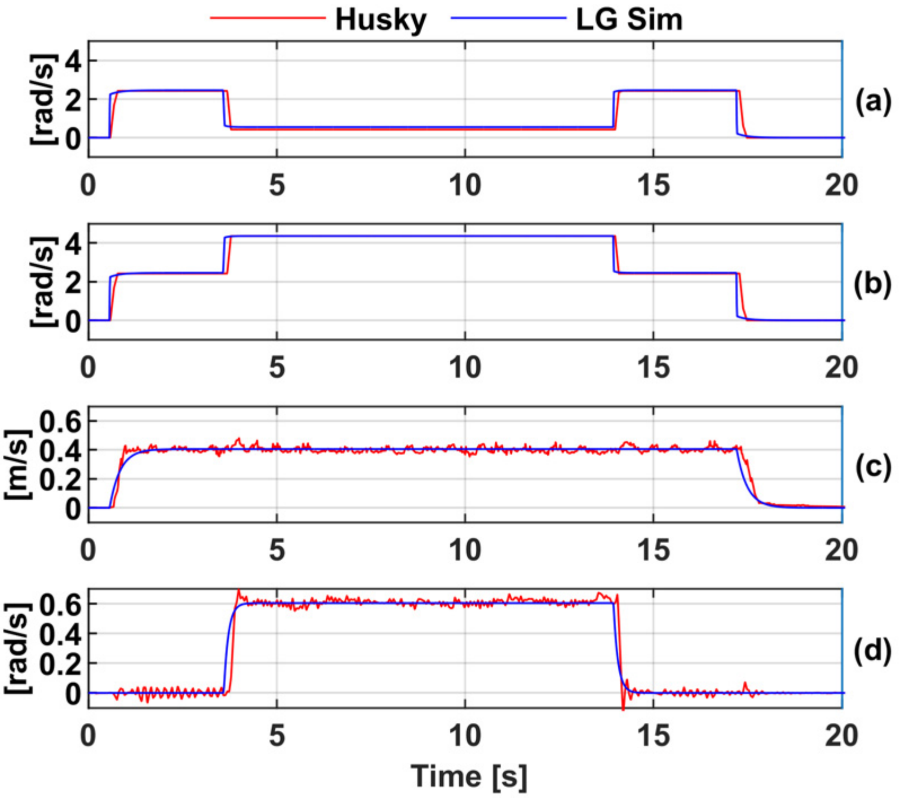


**Figure 13.** Response of (**a**) left- and (**b**) right-side wheel velocities and the (**c**) linear and (**d**) rotational velocities of the Husky LG model and the measured Husky data.

Figure 14 shows the trajectory response and the tracking error of the calibrated LG model in comparison to the trajectory of the physical robot and the Gazebo-simulated robot. As can be observed from this figure, the trajectory of the LG model closely agreed with the trajectory of the real robot, whereas the trajectory of the Gazebo simulation differed significantly, despite the same system inputs being used in the two simulation approaches. The bounds of the maximum tracking error of the LG-based simulation for this maneuver are

$$|\overline{X}| \leq 0.1397[m]|\overline{Y}| \leq 0.0819[m]$$

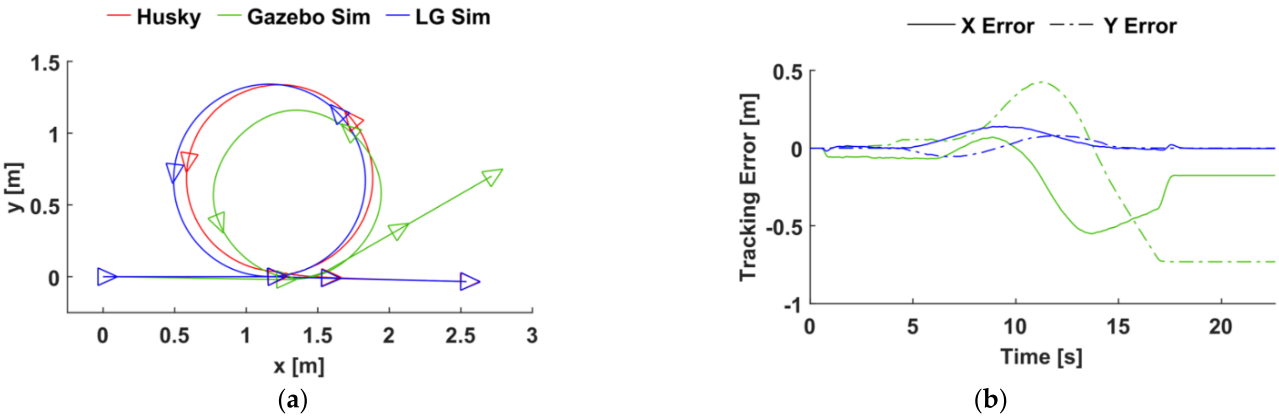


**Figure 14.** (**a**) Trajectory and (**b**) tracking error of the LG and Gazebo simulations and experimental data for the circle maneuver.

Figure 15 shows the trajectory response and the tracking error of the pre-calibrated LG model and the Gazebo simulations in comparison with the response of the physical robot for an S-bend maneuver. As can be observed from this figure, the trajectory of the LG model closely agreed with the trajectory of the physical robot while, again, the trajectory of the Gazebo simulation differed quite significantly. The bounds of the maximum tracking error of the LG-based simulation for this maneuver are

$$|\overline{X}| \leq 0.0352[m]|\overline{Y}| \leq 0.1027[m]$$

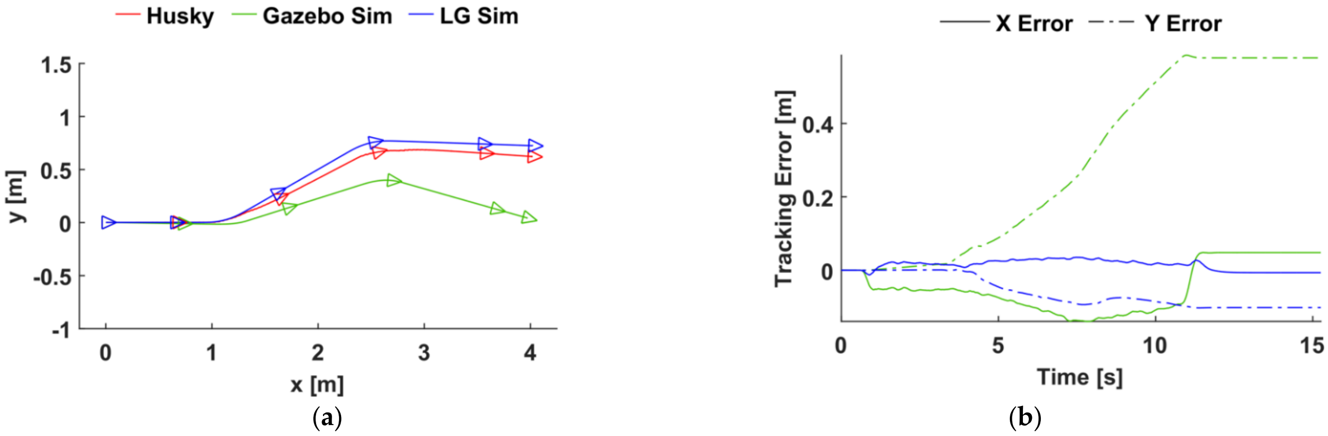


**Figure 15.** (**a**) Trajectory and (**b**) tracking error results of the LG and Gazebo simulations compared with experimental data for the S-bend maneuver.

Figure 16 presents the trajectory response and the tracking error of the pre-calibrated LG model and the Gazebo simulations in comparison with the response of the physical robot for an obstacle avoidance maneuver. Again, the trajectory of the LG model closely followed the trajectory of the real robot, whereas the trajectory of the Gazebo simulation differed significantly after the initial turn. The bounds of the maximum tracking error of the LG-based simulation for this maneuver are

$$|\overline{X}| \leq 0.0613[m]|\overline{Y}| \leq 0.0854[m]$$

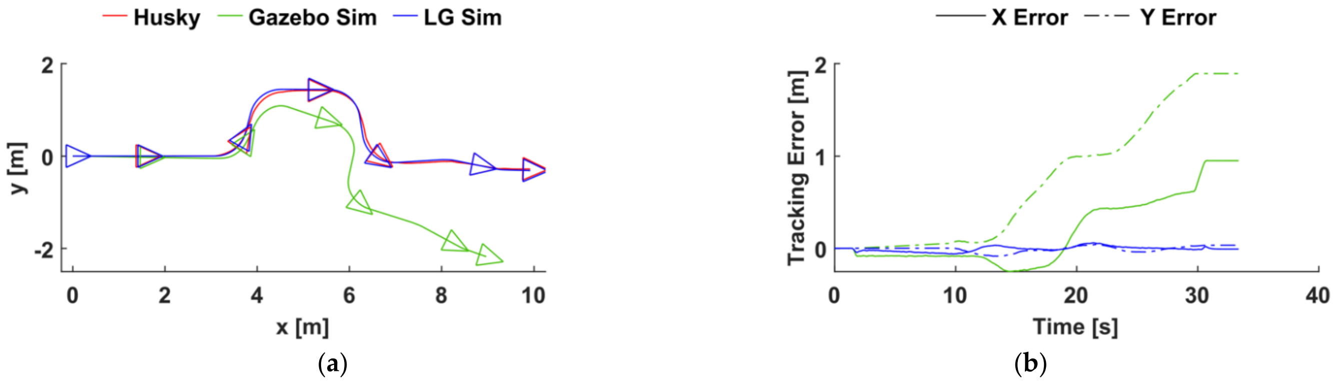


**Figure 16.** (**a**) Trajectory and (**b**) tracking error results of the LG and Gazebo simulations compared with experimental data for the obstacle avoidance maneuver.

In each of the three presented maneuvers, the calibrated LG model provided a significantly more accurate trajectory response than the Gazebo simulation. The primary cause of error for the Gazebo simulation was excessive skidding, which could be observed during rotational movements of the Husky during the simulation. For the circle maneuver, this excessive skidding resulted in a smaller, oval-shaped trajectory compared with the larger, circular trajectory of the LG model. Similarly, for the S-bend and obstacle avoidance maneuvers, the excessive skidding present in the Gazebo simulation resulted in skewing of the trajectory of the Gazebo robot. While some noticeable error was present in the LG model, it was much less than what was in the Gazebo simulations. The cause of this error was likely the complexity associated with modeling the dynamics of the interactions between the wheels of a skid-steer vehicle and the driving surface.

## 5. Conclusions

This paper presented a linear graph (LG)-based method of modeling the dynamics of a mobile robotic system, together with some background on the modeling approach of LG theory. The recently developed LGtheory MATLAB toolbox was utilized to automate the process of deriving the state-space model of a complex dynamic system such as the considered mobile robotic system. The genetic algorithm (GA) capabilities of the Global Optimization Toolbox were employed for estimating the unknown parameter values of the robot. The results of a comparison between a computer simulation of the state-space model for the mobile robot as generated by the LGtheory toolbox in comparison with an ROS- and Gazebo-based simulation and the experimental data collected while driving a physical Clearpath Husky mobile robot demonstrated and validated the accuracy of the developed modeling approach. The successful application of this modeling approach to a mobile robotic system, as demonstrated in the present work, provides further validation of both the proposed LG model and the custom software used to construct it.

The paper presented that LG theory is a desirable approach for achieving the “mechatronic purpose” in terms of features of integrated (or concurrent) multidisciplinary (or multi-domain or multi-physics) and unified approaches. It generated graphical models to conserve the natural structures of the physical systems. It can generate a unique analytical model in the state-space form directly. Meanwhile, it can be easily integrated with modern AIs and machine learning techniques in system modeling and design. The electro-mechanical model robot system in the case study showed its ability and broad application in other mechatronic applications.

**Author Contributions:** Conceptualization, H.L. and C.W.d.S.; methodology, H.L; software, E.M.; validation, H.L. and E.M.; formal analysis, E.M.; resources, H.L.; writing—original draft preparation, E.M.; writing—review and editing, H.L.; supervision, H.L. and C.W.d.S.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the NSERC Discovery Program, grant number RGPIN-2017-05762.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Paynter, H.M. *Analysis and Design of Engineering Systems*; The M.I.T. Press: Boston, MA, USA, 1961.
2. Sahoo, S.R.; Chiddarwar, S.S. Dynamic modelling and simulation of a three-Wheeled Omnidirectional Mobile Robot: Bond graph approach. In Proceedings of the Conference on Advances in Robotics (AIR 2017), New Delhi, India, 28 June–2 July 2017.
3. Sahoo, S.R.; Chiddarwar, S.S. Dynamic Modelling of Four Wheel Skid Mobile Robot by Unified Bond Graph Approach. In Proceedings of the 2016 International Conference on Robotics: Current Trends and Future Challenges, Thanjavur, India, 19–20 December 2016.
4. Huang, L.; Cheng, G.; Zhu, G.; Li, G. Development of a bond graph based model library for turbocharged diesel engines. *Energy* **2018**, *148*, 728–743. [CrossRef]
5. Niţu, C.; Grămescu, B.; Avram, M. Bond Graphs Modelling of Electromagnetic. *Int. J. Mechatron. Appl. Mech.* **2017**, *2017*, 118–123.
6. De Silva, C.W. *Modeling of Dynamic Systems—With Engineering Applications*; Taylor & Francis; CRC Press: Boca Raton, FL, USA, 2018.
7. Koenig, H.E.; Blackwell, W.A. Linear Graph Theory—A Fundamental Engineering Discipline. *IRE Trans. Educ.* **1960**, *3*, 42–49. [CrossRef]
8. Uren, K.R.; Van Schoor, G. State space model extraction of thermohydraulic systems—Part I: A linear graph approach. *Energy* **2013**, *61*, 368–380. [CrossRef]
9. Uren, K.R.; Van Schoor, G. State space model extraction of thermohydraulic systems—Part II: A linear graph approach applied to a Brayton cycle-based power conversion unit. *Energy* **2013**, *61*, 381–396. [CrossRef]
10. Dao, T.S.; McPhee, J. Dynamic modeling of electrochemical systems using linear graph theory. *J. Power Sources* **2011**, *196*, 10442–10454. [CrossRef]
11. McPhee, J. On the use of linear graph theory in multibody system dynamics. *Nonlinear Dyn.* **1996**, *9*, 73–90. [CrossRef]
12. McPhee, J.; Schmitke, C.; Redmond, S. Dynamic Modelling of Mechatronic Multibody Systems with Symbolic Computing and Linear Graph Theory. *Math. Comput. Model. Dyn. Syst.* **2004**, *10*, 1–23. [CrossRef]
13. Gamage, L.B.; de Silva, C.W.; Campos, R. Design Evolution of Mechatronic Systems Through Modeling, On-Line Monitoring, and Evolutionary Optimization. *Mechatronics* **2012**, *22*, 83–94. [CrossRef]
14. McCormick, E.; Lang, H.; de Silva, C.W. Research and Development of a Linear Graph-based MATLAB Toolbox. In Proceedings of the 14th International Conference on Computer Science & Education (ICCSE 2019), Toronto, ON, Canada, 19–21 August 2019; pp. 942–947.
15. Rowell, D.; Wormley, D.N. *System Dynamics: An Introduction*; Prentice Hall: Upper Saddle River, NJ, USA, 1997.
16. Karray, F.O.; de Silva, C.W. *Soft Computing and Intelligent Systems Design*; Addison Wesley: Harlow, UK, 2004.