

Research Article

Design of *sTetro*: A Modular, Reconfigurable, and Autonomous Staircase Cleaning Robot

Muhammad Ilyas,^{1,2} Shi Yuyao,¹ Rajesh Elara Mohan ,¹ Manojkumar Devarassu,¹ and Manivannan Kalimuthu^{1,3}

¹Engineering Product Development Pillar, Singapore University of Technology and Design (SUTD), Singapore 487372

²Department of Electrical Engineering, UET Lahore, NWL Campus, Lahore 54890, Pakistan

³School of Mechanical and Aerospace Engineering (MAE), NTU, Singapore 639798

Correspondence should be addressed to Rajesh Elara Mohan; rajeshelara@sutd.edu.sg

Received 4 April 2018; Accepted 30 May 2018; Published 24 July 2018

Academic Editor: Salvatore Pirozzi

Copyright © 2018 Muhammad Ilyas et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The mechanical, electrical, and autonomy aspects of designing a novel, modular, and reconfigurable cleaning robot, dubbed as *sTetro* (stair Tetro), are presented. The developed robotic platform uses a vertical conveyor mechanism to reconfigure itself and is capable of navigating over flat surfaces as well as staircases, thus significantly extending the automated cleaning capabilities as compared to conventional home cleaning robots. The mechanical design and system architecture are introduced first, followed by a detailed description of system modelling and controller design efforts in *sTetro*. An autonomy algorithm is also proposed for self-reconfiguration, locomotion, and autonomous navigation of *sTetro* in the controlled environment, for example, in homes/offices with a flat floor and a straight staircase. A staircase recognition algorithm is presented to distinguish between the surrounding environment and the stairs. The misalignment detection technique of the robot with a front staircase riser is also given, and a feedback from the IMU sensor for misalignment corrective measures is provided. The experiments performed with the *sTetro* robot demonstrated the efficacy and validity of the developed system models, control, and autonomy approaches.

1. Introduction

Due to a faster pace of life in most of the developed world, floor cleaning is often seen as a dull, dirty, laborious, time-consuming, and tedious job (Figure 1(a)) giving rise to the development of robotic products for handling the cleaning task autonomously. Such robotic platforms have given their vast potential by improving productivity in cleaning jobs in domestic and commercial settings and witnessed a steep rise over the last two decades [1]. It is estimated that between 2015 and 2018, about 25.2 million USD robotic cleaning units would be sold worldwide [2].

A number of successful products such as iRoomba, Neato XV-11, Samsung Powerbot, Bobsweep bobi, Miele scout, Moneual RYDIS, and Infinuvo CleanMate exist in the marketplace today [3]. These robots are generally characterized by nonmodular/fixed circular morphology capable of

autonomously mapping its environment using on-board sensors and navigating around the defined floor space to clean the smooth surface environment efficiently. The limitation of these conventional robots is that they are applicable on smooth floor-like surfaces only and are unable to clean staircases which are an essential part of most of homes/offices as shown in Figure 1(b).

Numerous research literature deal with different aspects of floor cleaning robots such as the mechanism design [3–5], autonomy [6, 7], human-robot interaction studies [8, 9], multirobot teams [10, 11], and benchmarking strategies [12, 13]. Even though there exists such literature demonstrating the benefits of floor cleaning robots, the conventional floor cleaning robots suffer from serious performance issues that curtail their full potential dexterousness. One major factor attributing to their performance loss is their inability to access staircases that form the integral part of almost every



FIGURE 1: (a) Manual staircase cleaning. (b) Conventional home cleaning robots.

built infrastructure. Considering the crucial role staircases play and their permanent presence even after the advent of lifts and escalators virtually in every multistory building, the subject of cleaning staircases has received little attention from roboticists. One viable approach to overcome this bottleneck is to design next-generation cleaning robots that are able to reconfigure themselves between floor and staircase cleaning modes, thereby maximizing their dexterous task performance. A number of design mechanisms towards realizing staircase climbing robots have been proposed and validated [14–18]. However, these robots target mostly search, rescue, and security applications using design principles and mechanisms that are not optimal for cleaning tasks. A very limited research effort has been made towards the design of staircase cleaning robots. An autonomous cleaning robot is proposed in [19] capable of accessing both flat floors and staircases. This robot consists of a rectangular-shaped body frame with L-shaped legs on both sides of the frame and moves forward and climbs stairs by rotating the body so that the top and bottom sides of it may be reversed using L-shaped legs. Megalingam et al. in [20] present various design considerations for vertical climbing robots to be used in staircase cleaning. The work describes a case study with the design of one such robot that can function in a 3D space. However, these robotic platforms that target staircase cleaning are complex and suffer from serious design and performance issues.

The *sTetro* (a staircase cleaning robot) is a novel modular reconfigurable cleaning robot, which uses a vertical conveyor belt mechanism and is capable of navigating over flat surfaces (e.g., floor) as well as climbing staircases, thus significantly extending the cleaning capabilities of home cleaning robots. The working principle concept of *sTetro* is borrowed from the Tetris Tiling Theory [21]. The *sTetro*'s body consists of three cuboids connected with two sliders attached to each side of the central cuboid. The hollow space inside each parallelepiped block encloses the electronic units, vacuum/suction tools, and sweeping mechanism. The blocks have a modular design, which allows reusing many pieces in the system. An intelligent application

allows the user to control the robot to start and stop, navigate to follow a defined path, and change the operational configuration, as desired. The reconfiguration mechanism allows the *sTetro* robot to efficiently clean both the floor and the stairs at one time without user intervention, which is almost impossible in presently available home cleaning robots in the market.

In this paper, the mechanical design and system architecture of *sTetro* are introduced, followed by a detailed description of system modelling and controller design efforts. An autonomous motion algorithm is also proposed and validated with real robot data. To keep a robot aligned with the front riser of the staircase, an IMU sensor is used for heading change tracking and a feedback signal is a feedback to control for misalignment correction, during left/right movement and climbing the stair steps. In fully autonomous mode operation, the robot must be able to find/recognize the staircase. Many algorithms exist which try to recognize stairs with ultrasonic sensors [22], RGBD camera [23–25], stereo camera [26], and/or lidar [27]. These algorithms are complex and use already existing techniques for recognition. Camera-based recognition techniques suffer from lighting conditions as well. In this work, we have made use of a 2D lidar fixed in a vertical configuration to distinguish between the staircase and the surrounding environment. A staircase detection algorithm is developed using 2D lidar scans, which are matched with precaptured reference scan data.

Experiments performed with the *sTetro* robot demonstrated the viability of the original design concepts, system model autonomy, and control approaches.

2. *sTetro*: System Architecture

2.1. Mechanical Design. In Figure 2(a), the proposed prototype of the staircase cleaning robot is shown and a typical operation environment of *sTetro* is shown in Figure 2(b). A desktop 3D printer was used to print the entire robot's mechanical parts with polylactic acid (PLA) material. With many prototyping methods available, the *Fused Deposition*



FIGURE 2: (a) *sTetro* images. (b) Typical working environment of *sTetro* (stairs with treads and risers).

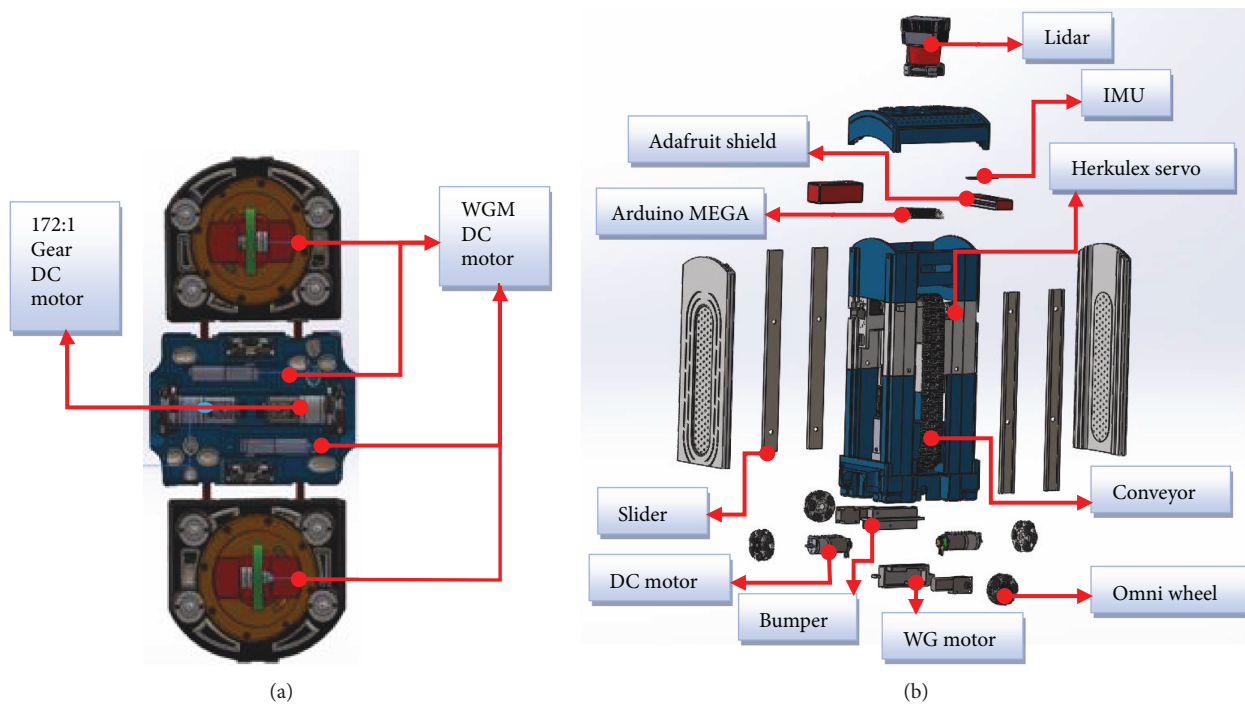


FIGURE 3: (a) Bottom view. (b) Exploded view of *sTetro*.





Melting (FDM) method was used in the development of the *sTetro* robot which has an elastic modulus of 3.5 GPa and a density value of 1.3 g/cm^3 .

Figure 3 presents the bottom view and an exploded view of the *sTetro* robot. The *sTetro* platform has three segments/blocks with two metal sliders attached to each side of the middle block. The components in each block are distributed accordingly to maintain the block's weight which ensures the centre of gravity (CoG) of the robot during its locomotion and self-reconfiguration.

2.2. Electrical Design. The *sTetro* robot consists of six DC motors; among them, four (M1–M4) are Worm Gear DC motors, and two (M5, M6) are Pololu DC motor. It also has

four high torque servo motors (Herkulex 0201: SM1 to SM4). The motors M1 to M4 operate at 11.1 V and draw an average current of 800 mA with power rating 1600 mAh to deliver a nominal torque of 3 kg cm. The Pololu DC motor (M5, M6) has more torque of about 14.5 kg cm and has a gear ratio of 172:1. Four Omni-directional wheels are used in *sTetro*, with a diameter of 48 mm, to navigate the robot in planar directions (X, Y plane). Two of these Omni wheels (forward wheel) are driven by Pololu motors (M5, M6), and the side Omni wheels are driven by Worm Gear motors (M2, M3). In the 1st and 3rd blocks, DC motors M1 and M4 are installed to control the motion of monowheels (H-wheels), and Herkulex servo motors, SM3 and SM4, are used to control the direction of the 1st and 3rd block mono

TABLE 1: Mechanical components with specifications.

Component	Specs.	Remarks
Block-1 and block-3 Block-2	$L = 147 \text{ mm}$, $W = 172 \text{ mm}$, $H = 390 \text{ mm}$, weight = 3 kg $L = 135 \text{ mm}$, $W = 172 \text{ mm}$, $H = 430 \text{ mm}$, weight = 2 kg	Total weight = 7 kg
4 × Omni-directional wheels	Dia = 48 mm	
2 × conveyor belt mechanism	$L = 220 \text{ mm}$, $W = 36 \text{ mm}$	
2 × contact switch	SPDT	
6 × bumper (long)/RB-Nex-31		

wheels. At the 2nd block, two servo motors (Herkulex: SM1, SM2) are attached to lift all three blocks using a vertical conveyor belt mechanism (Figure 3(b)). Six long bump sensors are attached on the left/right sides of the robot to detect side boundaries of the stair, and two time of flight (ToF) sensors are attached on the bottom of block-1 and block-2 to detect touch down on tread. A 3rd ToF sensor is attached in front of block-3 to detect front riser touching during reconfiguration. The more detailed internal views of *sTetro* are given in Figure 3.

The mechanical and electrical parts/components used to construct this robot are listed in Tables 1 and 2, respectively, for clarity and easy understanding of *sTetro* model.

3. Locomotion and Reconfiguration Control of *sTetro*

The sequence of sensing and control actions enables the *sTetro* robot to reconfigure itself and move forward, left/right wards and upwards. In the previous prototype [28], motion and reconfiguration actions were controlled by a human operator by sending control commands through a remote control, but in the current version, we have automated most of these actions. Different sensors are installed to accomplish the self-reconfiguration and locomotion task. To make the locomotion and self-reconfiguration of *sTetro* automatic, simple, low-cost, low-weight, and power-efficient sensors have been used in this project. As it is a slow-moving wheeled robot to physically interact/negotiate with its environment (e.g., walls/stairs and human), the contact sensors are used in the design due to the robot's slow motion and low momentum. Moreover, physical contact sensors are unaffected with environmental conditions, for example, light conditions and target material type, and are more accurate than electronic sensors. Contact sensors react only when they have

physical touch with the surrounding environment and have a zero limiting range. Six long bump sensors (BM1~6) are installed on the left and right sides of the robot to detect the side boundaries of the staircase. A contact switch (SW0) is installed in the front of block-1. Two touch sensors (SW2 and SW3) of the SPDT type are fixed on the front face of block-2 to detect the front riser and smooth sliding motion against the front riser. Two ToF1 and ToF2 sensors are fixed in the bottom of each block-1 and block-2 to detect the touchdown of these blocks on the tread of the stair. The third ToF3 sensor is attached in the front of block-3 to detect the front riser. The navigation sensors installed on *sTetro* are shown in Figure 4.

When the Herkulex-1 (SM1) actuates clockwise, it initiates the upward motion of block-1, and when SM1 actuates in the counter clock direction, it enables the downward motion of block-1. Similarly, when Herkulex-2 (SM2) actuates in the clockwise/counterclockwise direction, it enables the upward/downward motion of the 3rd block. When SM1 actuates in a clockwise direction and SM2 actuates in the counterclockwise direction, the upward and downward motions of the block-2 are achieved, respectively.

An Arduino MEGA 2560-based microcontroller and an Adafruit DC motor shield are used in the *sTetro* robot to achieve its locomotion and reconfiguration. This unit is placed inside the middle block. An IMU (PMU-6050) is placed in block-2 as well to detect the heading angle of the robot and to be kept aligned with the front riser of the staircase. The microcontroller controls all the 6 × DC motors by sending PWM values through the DC motor shield. The unit also controls the 4 × servo motors to control its position and angle during locomotion. The Adafruit DC motor shield has 4 channels, MD1–MD4, which are used to address each DC motor. The entire robot is powered using a 3 × 7.4 V (2100 mAh) and 1 × 11.1 V lithium polymer (1600 mAh)

TABLE 2: Electrical components with specifications.

Component/model	Specs.	Remarks
4 × DC motors/WGM34	No load RPM: 100; stall torque: 3.5 kg cm; rated voltage: 12 VDC; no load current: 0.08 A max	
2 × Pololu DC motors/172: 1 gear ratio	25D × 56 L mm; no load RPM: 34 RPM; stall torque: 14.5 kg cm; rated voltage: 6 VDC; stall current: 2400 MA max	
4 × servo motors (SM1 and SM2)/ Herkulex Drs-0201	Operating angle: 320°; stall torque: 24 kg cm; rated voltage: 7.4 VDC	
1 × microcontroller	Arduino MEGA 2560	
1 × Adafruit DC motor shield	L293D motor shield	
1 × Bluetooth module	HC-06, 2.4 GHz	
4 × batteries (BT1~BT4): BT1, BT2 (block-2); BT3 (block-1), BT4 (block-3),	BT1: 11.1 V Li-po with 1600 mAh (to power 6 DC motors and Arduino) BT2: 7.4 V Li-Po with 2100 mAh (to power Herkulex: SM1, SM2) BT3: 7.4 V Li-Po with 2100 mAh (Ttzo power Herkulex: SM3) BT4: 7.4 V Li-Po with 2100 mAh (to power Herkulex: SM4)	
1 × IMU	MPU-6050 6 DoF motion sensor	
3 × VL6180X, proximity sensor	Dia: 4.8 × 2.8 × 1.0 mm; range: 0 to 100 mm; 2.6 to 3.0 V; 1.7 mA	
1 × solid-state 2D lidar/LS02A (proposed)	FOV: 86°; distance accuracy: 1.5%; distance range: 0.1–4 m; scan freq.: 10 Hz; angular resolution: 1°/0.5°	

battery power. A voltage regulator is used to distribute power to two servo motors. Additionally, a proportional-integrative-derivative (PID) controller has been implemented to control the position and angle of the robot with feedback from the IMU sensor. To control the locomotion and self-reconfiguration modes of the developed robot, we have designed an Android application using Android Studio in Java. A Bluetooth module (HC-06) with a band frequency of 2.4 GHz is used to communicate with the robot. The system architecture of the sTetro robot is illustrated in Figure 5.

4. sTetro: System Modeling

4.1. Torque of Conveyor Motor. The modelling of the vertical conveyor system that forms the core of the sTetro robot was the critical component system design. The analysis is done on the relationship between the torque required and the mass of each block. As shown in Figure 6(a), the forward locomotion requires the vertical motion of the blocks which is driven by the motor marked as the red dot. The torque of the conveyor motor and the mass of each block determine whether it is feasible for the

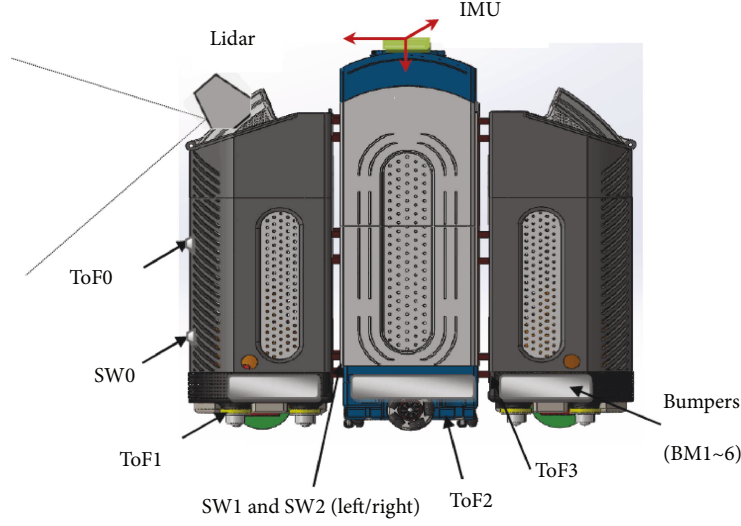


FIGURE 4: Navigation sensors installed on *sTetro*.

conveyor motor, and the mass of each block determines whether it is feasible for the robot to reconfigure itself and complete the staircase climbing locomotion. The weights of each block are represented by m_1 , m_2 , and m_3 . The black blocks in the middle are the motors, and the red geometry is the position of the conveyor. Figure 6(b) shows the schematic diagram and the parameter used in the conveyor torque analysis. In Figure 6(b), point A refers to the position of the motor, W_1 and $F_{1,req.}$ are the weight and the force required to lift up block-1 while L_{W1} and L_f refer to their corresponding moment, and N_{11} , N_{12} , N_{13} , and N_{14} are the normal forces exerted on block-1 while d refers to the distance between the mounting points.

The dynamic model of *sTetro* is formulated below. The summation of moments at point A on the linear guide which carries block-1 (m_1) is given as

$$\begin{aligned}
 \sum M_A &= 0, \\
 W_1 L_{W1} + F_{m1,req.} L_{f1} - (N_{11} + N_{12} + N_{13} + N_{14}) \frac{d}{2} &= 0, \\
 N_{11} + N_{12} + N_{13} + N_{14} &= \frac{2W_1 L_{W1} + 2F_{m1,req.} L_{f1}}{d}, \\
 \sum F_y &= F_{m1,req.} - W_1 - F_{f1} = m_1 \ddot{y}, \\
 F_{m1,req.} - W_1 - \mu(N_{11} + N_{12} + N_{13} + N_{14}) &= m_1 \ddot{y}, \\
 F_{m1,req.} - W_1 - \mu \left(\frac{2W_1 L_{W1} + 2F_{m1,req.} L_{f1}}{d} \right) &= m_1 \ddot{y}, \\
 F_{m1,req.} &= \frac{m_1 \ddot{y} d + W_1 (d + 2\mu L_{W1})}{d - 2\mu L_{f1}}.
 \end{aligned} \tag{1}$$

Hence, simplifying the equations above gives out

$$\begin{aligned}
 \sum M_A &= F_{m2,req.} L_f + (N_{21} + N_{22} + N_{23} + N_{24}) 0.5d \\
 &\quad - 0.5W_2 L_{W2} = 0, \\
 N_{21} + N_{22} + N_{23} + N_{24} &= \frac{W_2 L_{W2} - 2F_{m2,req.} L_{f2}}{d}, \\
 \sum F_y &= F_{m2,req.} - 0.5W_2 - F_{f2} = 0.5m_2 \ddot{y}, \\
 F_{m2,req.} - 0.5W_2 - \mu(N_{21} + N_{22} + N_{23} + N_{24}) &= 0.5m_2 \ddot{y}, \\
 F_{m2,req.} - 0.5W_2 - \mu \left(\frac{W_2 L_{W2} - 2F_{m2,req.} L_f}{d} \right) &= 0.5m_2 \ddot{y}, \\
 F_{m2,req.} &= \frac{0.5m_2 \ddot{y} d + W_2 (0.5d + \mu L_{W2})}{d + 2\mu L_f}.
 \end{aligned} \tag{2}$$

4.2. Centre of Gravity (CoG) Analysis. The purpose of CoG analysis is to verify the stability of the robot when not all wheels are in contact with the ground during ascending and descending staircases. This section will deal with the stability calculation of the robot under the typical static scenarios. Figure 7(a) shows parameters used in the calculation and the free-body diagram (FBD) when block ① is lifted up. The *sTetro* robot consists of three blocks which are labelled as ①, ②, and ③. As the normal forces exerted on the robot are on the x-z plane, they are labelled in the bottom diagram of Figure 7(a). During the calculation, the whole unit is assumed to be static and stable. The stability is verified by identifying the direction of the normal forces.

According to the FBD, the force and moment equation of the whole unit can be expressed as

$$\begin{aligned}
 \sum \vec{F} &= 0, \quad \therefore (m_1 + m_2 + m_3)g = \sum_{i=1}^8 N_i, \\
 \sum \vec{M} &= 0.
 \end{aligned} \tag{3}$$

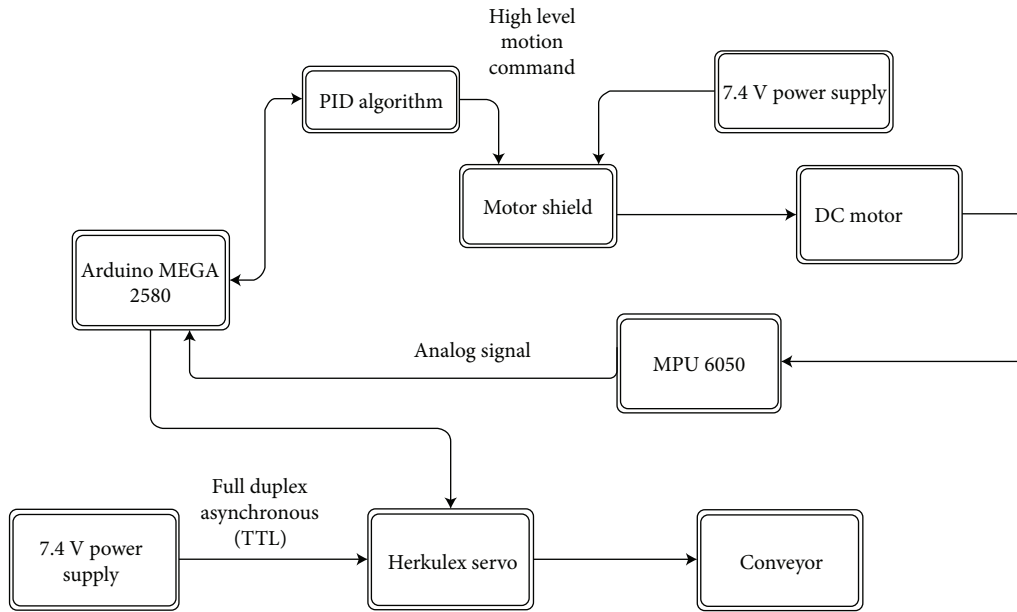


FIGURE 5: System architecture of *sTetro*.

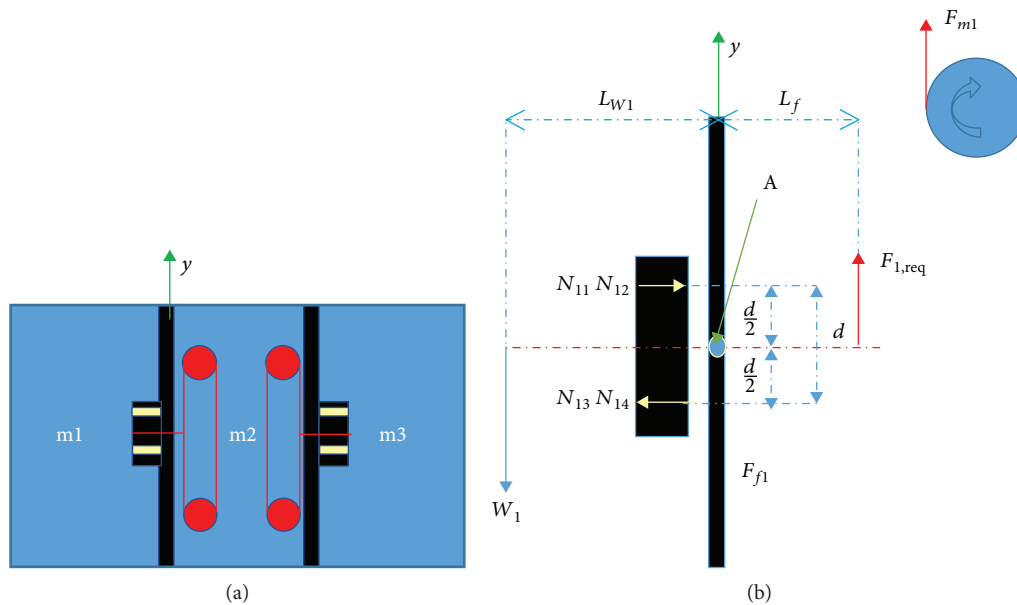


FIGURE 6: Parameters used in conveyor torque analysis.

Similarly, the equations of block ①, ②, and ③ are written below. For block ①, parameter a is an unknown length depending on the height of block ①.

$$\begin{aligned} \sum \vec{F}_x = 0, & \quad \therefore F_{r1,2} = F_{r2,1}, \\ \sum \vec{F}_y = 0, & \quad \therefore m_1 g = V_{r2,1}, \\ \sum \vec{M} = 0, & \quad \therefore \frac{m_1 g L}{2} = m_{r2,1} = \frac{3F_{r2,1}a}{4} - \frac{3F_{r1,2}a}{4}. \end{aligned} \quad (4)$$

For block ②,

$$\begin{aligned} \sum \vec{F} = 0, & \quad \therefore F_{r1,2} + F_{r2,3} = F_{r2,1} + F_{r3,2}, \\ \sum \vec{F}_y = 0, & \quad \therefore m_2 g + V_{r1,2} - V_{r3,2} = \sum_{i=5}^8 N_i, \quad V_{r1,2} = V_{r2,1}, \\ \sum \vec{M} = 0, & \quad \therefore 0.5m_2 g L - N_5 L - 0.5(N_6 + N_8)L - 0.5V_{r3,2}L \\ & \quad = M_{r3,2} - M_{r2,1}, \\ M_{r3,2} & = \frac{3F_{r3,2}H}{4} - \frac{3F_{r2,3}H}{4}. \end{aligned} \quad (5)$$

Similarly, for block ③,

$$\begin{aligned}
\sum \vec{F}_x &= 0, \quad \therefore F_{r2,3} = F_{r3,2}, \\
\sum \vec{F}_y &= 0, \quad \therefore m_3g + V_{r2,3} = \sum_{i=1}^4 N_i, \\
\sum \vec{M} &= 0, \quad \therefore \frac{m_3gL}{2} - N_1L - \frac{(N_2 + N_4)L}{2} = -M_{r2,3}, \\
M_{r2,3} &= M_{r3,2}.
\end{aligned} \tag{6}$$

For blocks ② and ③ as a whole,

$$\begin{aligned}
\sum \vec{F}_x &= 0, \quad \therefore F_{r1,2} = F_{r2,1}, \\
\sum \vec{F}_y &= 0, \quad \therefore (m_2 + m_3)g + V_{r1,2} = \sum_{i=1}^8 N_i, \\
\sum \vec{M} &= 0, \quad \therefore \frac{m_1gL}{2} + V_{r1,2}L + M_{r1,2} + N_1L + \frac{(N_2 + N_4)L}{2} \\
&= \frac{m_3gL}{2} + \frac{(N_6 + N_8)L}{2} + N_7L.
\end{aligned} \tag{7}$$

For the robot to be stable, the constraints are set to be $N_i \geq 0, i = \{1, 2, 3, 4, 5, 6, 7, 8\}$. The extreme case is when all the normal forces are equal to zero; that is, $m_1g = 4(m_2g + m_3g)$. By substituting the dimensions from the design, the constraints are fulfilled. Therefore, the whole robot is in the stable condition.

4.3. PID Control Approach. The implementation of a PID controller in this system is necessary as there are always multiple driving and driven wheels working simultaneously. Any asynchronization of three blocks will cause it to deviate from the original orientation. The mathematical modelling and simulated response were done in [28]. The block diagram of the control system is shown in Figure 7(b). The overall transfer function of the system is given as (complete derivation of the transfer function is given in [28])

$$\frac{\Phi(s)}{E_2(s) - E_1(s)} = \frac{r}{y} \frac{2K}{bR + KK_b((J_w R/(bR + KK_b))s + 1)s}. \tag{8}$$

Substitute the constants into the equation,

$$\frac{\Phi(s)}{E_2(s) - E_1(s)} = \frac{0.03}{0.3} \frac{2 \times 0.955}{b1.82 \times 10^{-5} \times 6.324 + 0.955 \times 6.29 \times ((1.707 \times 10^{-4}/6.26)s + 1)s}. \tag{9}$$

By using *SISOTOOL* in *MATLAB*, the simulation of the step and impulse response are shown in the figures below with corresponding bode diagram and values of gains. The optimized transfer function of the controller is found as

$$G_c(s) = \frac{0.057728}{s} + 20.78 + 161.63s. \tag{10}$$

Therefore, the proportional gain is 20.78, the integral gain is 0.057728, and the derivative gain is 161.63. Under the control of the above controller, Figures 8 and 9 show the step and impulse response of this system.

The simulated response of the system to unit step input shows that the final value approaches one with zero steady-state error. Similarly, the impulse response of the system simulates the ability of handling impulse disturbance from the surrounding because of the implementation of the integration term. Both of the poles of controlled systems are placed at the turning point on the magnitude bode diagram to reduce the settling time.

5. Autonomy Algorithm of sTetro Robot

To make the locomotion and self-reconfiguration of sTetro automatic, simple, low-cost, low-weight, and power-efficient sensors have been used in this project. A time of flight (ToF) lidar is used for distance measurement to obstacle/stair, and a MEMS IMU is used for misalignment detection from the front riser of staircase. In addition, two

ToFs are used in the bottom of block-1 and block-2 to detect touching down. In order to achieve autonomy in sTetro, the following sensors/components are used:

- (i) 1 × contact switch (SW0) in the front side of block-1 (to detect the front riser of the stair)
- (ii) 2 × contact switches (SW1 and SW2) at both sides of block-2 (to detect the front riser of the stair and for the smooth motion against riser)
- (iii) 6 × side bumpers (B1–6) at both sides of block-1~block-3 (to detect the end of sides of the staircase)
- (iv) 2 × ToF sensors (ToF1 and ToF2) at the bottom of block-1 and block-2 (to detect touchdown on tread)
- (v) 1 × ToF sensor (ToF3) in the front side of block-3 (to detect front riser of the stair)
- (vi) 1 × IMU fitted in block-2 (to detect misalignment of the robot)
- (vii) 1 × 2D lidar for staircase recognition

5.1. Automatic Staircase Recognition. A 2D lidar is used, in vertical scanning configuration, to capture the staircase structure scans in the working environment. As in this application, the staircase has a unique feature as compared to surrounding walls and home features (table, chairs, beds, etc.); hence, by exploiting this distinguishing shape of the

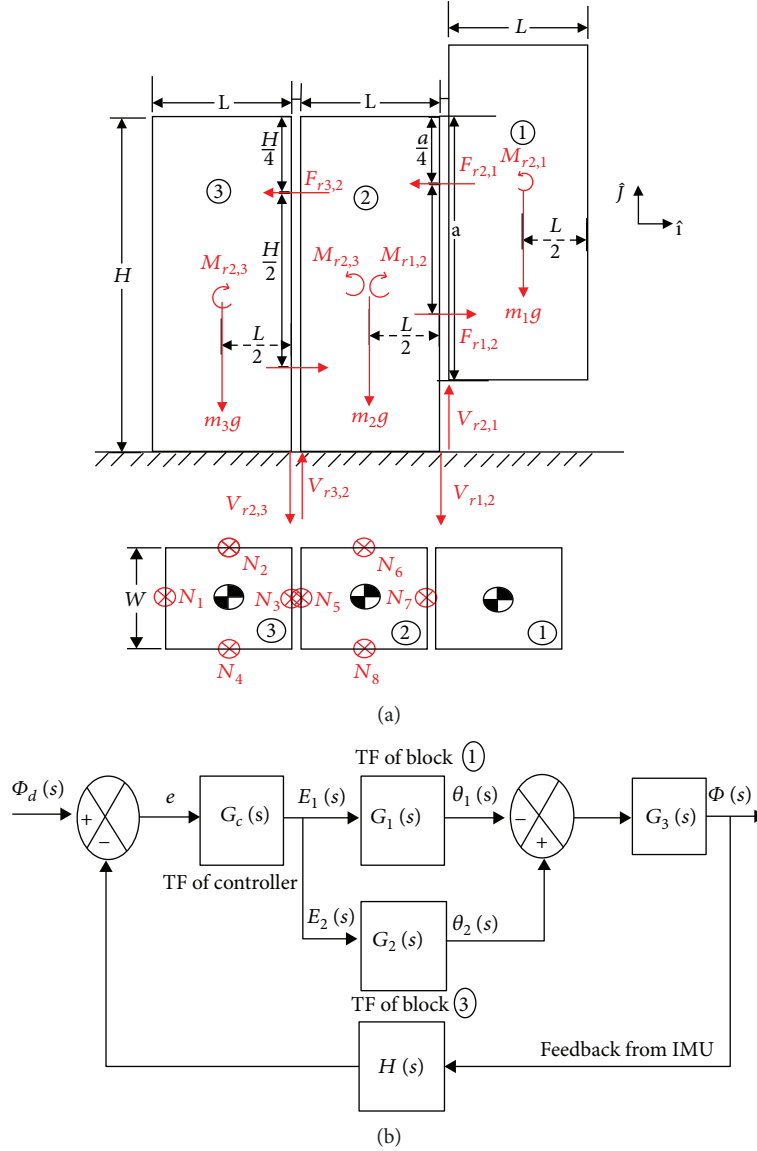


FIGURE 7: (a) Schematic of FBD of *sTetro* robot. (b) Control scheme.

staircase, we have developed a simple algorithm to initialize the staircase cleaning operation by *sTetro*.

The *automated staircase recognition algorithm* starts with a capture of a reference scan (S_R) with the lidar installed on *sTetro* in the vertical direction (Figure 10(a)). Due to this particular placement of the lidar, the staircase structure will appear as steps (riser and tread) when lidar ranges are obtained in a scan. While capturing the reference scan, the *sTetro*/lidar is facing (almost perfectly) to the staircase, and thus, the scan range obtained looks like a staircase pattern. After getting and storing S_R , we place *sTetro* in any arbitrary orientation in front of the stairs, and the robot starts taking rotations to scan its environment. At each angular position, the measured scan (S_M) is compared, by calculating the Euclidean norm (also known as the *square root of the sum of squares of differences*), with already stored reference scan and the one which gives the minimum Euclidean distance (D_i) is assumed to be the

closest angular position with the reference orientation (facing staircase).

$$D_i = \sqrt{\sum_i (S_R - S_{M_i})^2}, \quad (11)$$

where D_i is the Euclidean norm.

Before comparing the two scans, we also have designed a simple outlier filter to reject the spurious points. The technique used is simple. In the range data, the difference between consecutive lidar points is calculated; if this difference is larger than some threshold value, that particular point in the scan is considered a spurious point and is replaced with the mean value of the previous point and this spurious point. This enables smoother data points in a particular scan and avoids any overestimate of the Euclidean norm value, as shown in Figure 10(b). A pseudocode of this algorithm is provided in Pseudocode 1 below.

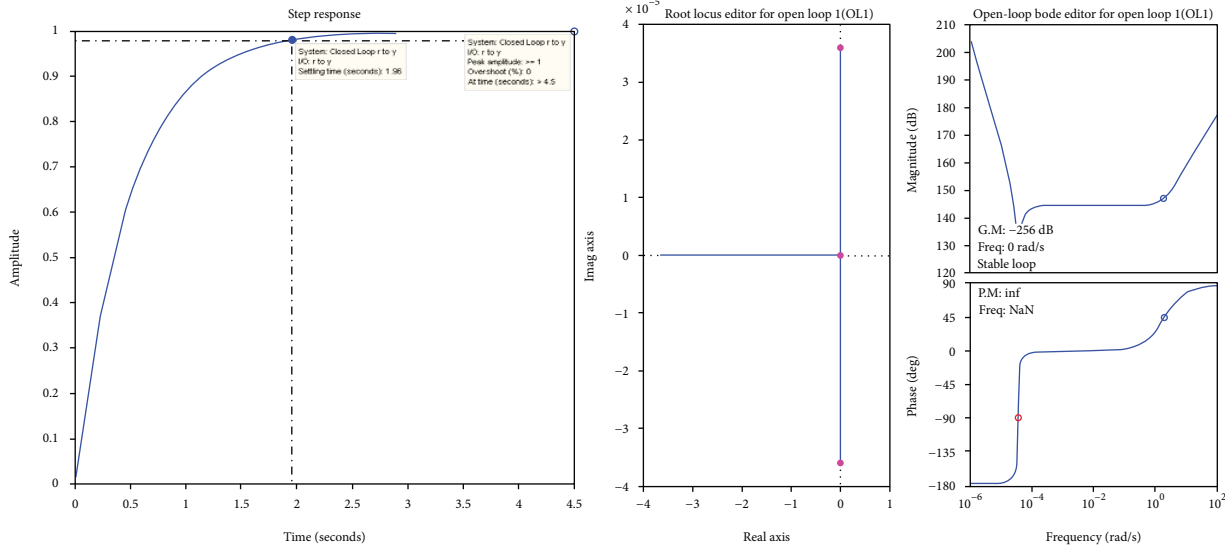


FIGURE 8: Step response of the system with bode plot.

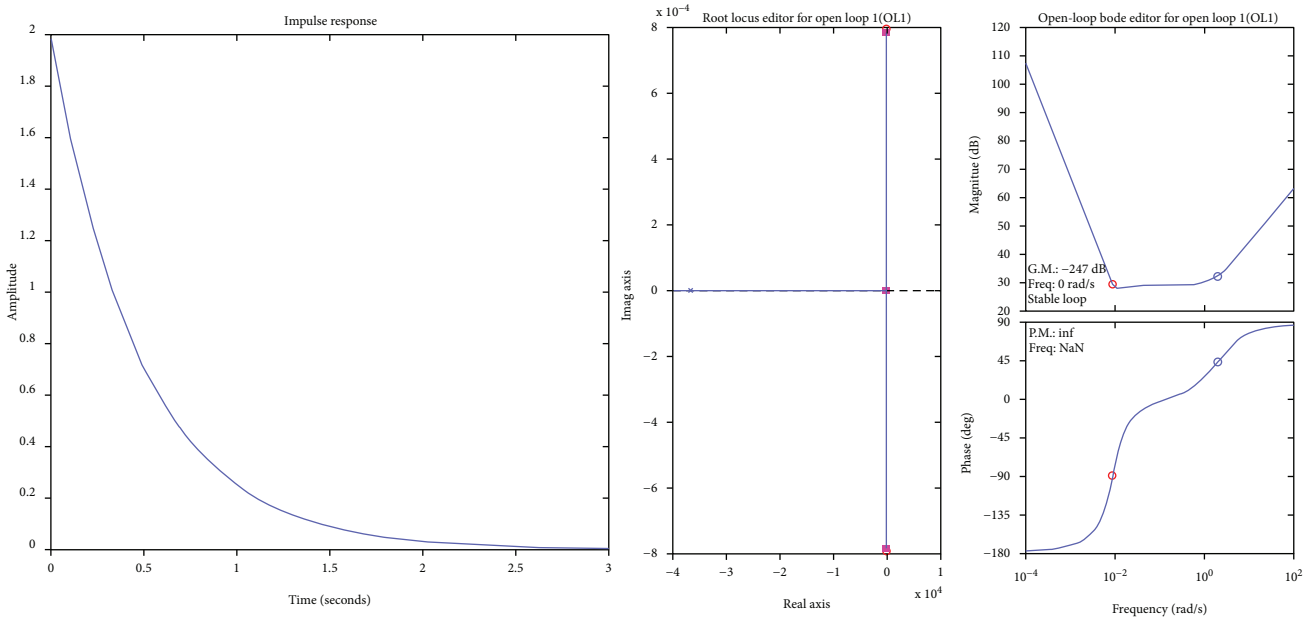


FIGURE 9: Impulse response of the system with bode plot.

5.2. *Misalignment Correction Technique.* During operation, the *sTetro* robot must be aligned with the staircase riser so that it can cover all tread area for cleaning purposes. If some misalignment occurs uncorrected, the robot may fall back from the narrow tread’s area. In order to detect misalignment with the front riser, an attitude and heading reference system (AHRS) has been used in this work. The AHRS consists of inertial IMU sensor ($3 \times$ gyros, $3 \times$ accelerometers) on three orthogonal axes that provide attitude information (roll, pitch, and yaw/heading) for any moving platform on which the sensor is installed. When the robot touches the riser in the first step and is aligned with the staircase, the yaw angle at that position is taken as the reference heading, and any

subsequent changes from this reference heading angle are taken as misalignments (with some tolerance/threshold). The AHRS algorithm is able to detect such misalignments very reliably and gives feedback to the control system for corrective measures.

For the *sTetro* to move and reconfigure itself autonomously, the *high-level description* (pseudocode) of the autonomy algorithm is given in Pseudocode 2 below:

6. Experimental Results

Experiments are performed with the *sTetro* platform to demonstrate the efficacy and validity of the developed algorithms

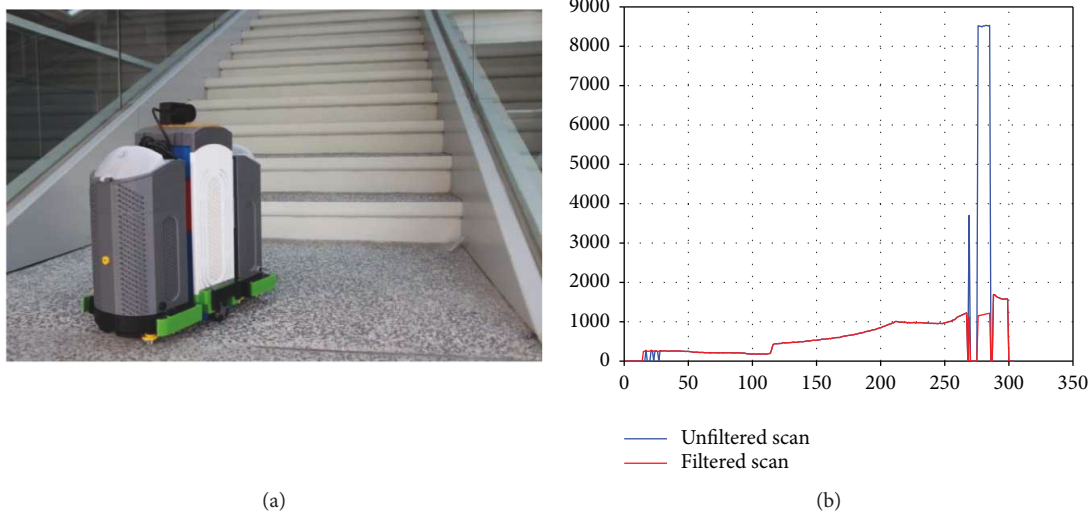


FIGURE 10: (a) 2D lidar installed on *sTetro*. (b) Range data with an outlier rejection filter effect.

```

Automatic Staircase Recognition Algorithm in sTetro:
Capture reference scan ( $S_R$ ); // by placing robot in front of stairs such that lidar rays
//are perpendicular to the stairs.
OutLierFilter( $S_R$ ); // apply OutLierFilter() to reject any superiors points.
Save  $S_R$ ;
While (Euclidean Norm is Not Min.)
    TURN in-place; // rotate CCW
    Capture measured scan ( $S_{M_i}$ );
    OutLierFilter( $S_{M_i}$ );
     $D_i \leftarrow \text{CalcEuclideanNorm}(S_R, S_{M_i});$  //calculate Euclidean Norm between ref. scan and
//measured scan.
    Store  $D_i$ ;
    If ( $D_i < D_{i-1}$ )
        Continue TURN in-place; // continue rotating till Euclidean dist. is min.
    end if
endWhile
STOP Turning.

```

PSEUDOCODE 1: Staircase Recognition.

of the control system, the torque generated by the conveyor motor as well as autonomous mode operation. This section presents the experimental results in detail.

6.1. PID Controller Experiments. The first set of experiments focused on testing the response of the robot during the side-way motion without any disturbance. The unit is switched on and commanded to move in the direction that is perpendicular to its body as shown in Figure 11(a). Initially, the accelerometer is calibrated as the zero degree. Although all motors belong to the same model, the levelling issue arising from the three blocks resulted in the slippage in wheels and thereby deviation in the direction of motion. The values of three gains used in the experiments are based on the theoretical analysis. Figure 11(b) shows the time plot of the angle it deviates (φ)

during the side-way motion without disturbance. The plot shows that it tends to deviate towards the positive side. The suspected reason is the wear-off of the motor shafts which are 3D printed in PLA material. In addition, the weight of the robot could have contributed to the loose connection between the shaft and the motor, resulting in further deformation of the wheel and deviation in the motion.

The second set of the experiment involved the study of the step response during the side-way motion. After the calibration, the robot is placed with a nonzero initial angle and commanded to move sideways. The expected angle is set as zero as a step input to the system. Figure 12(a) shows the time plot of its deviation angle. Lastly, the experiment involved the study of the impulse response during the side-way motion. Instead of placing the stopper in front, the impulse response

Pseudocode of Autonomy Algorithm in sTetro:

```

// Initialization Start
While (staircase Not detected)           // using staircase recognition algorithm.
    TURN in-place;                       // rotate CCW
endWhile
    STOP Turning;
Initialization ends
//Step#1;
If (ToF0_range < 1500mm)                 // continue staircase climbing until ToF0 > 1.5 m.
// Forward motion.
While (ToF0_range > 5mm)                 // move fwd until ToF0 measure < 5 mm.
    MOVE FWD;
endWhile
STOP Moving fwd;
LIFT Block-1 to ~18 cm;                  // determined by mechanical lock.
While (SW1_OFF or SW2_OFF)              // move fwd until SW1 or SW2 is ON.
    MOVE FWD;
endWhile
STOP Moving fwd;
While (ToF1_Range_Down < 1)              // drop Block_1 until ToF1 senses < 1cm range.
    DROP Block-1;
endWhile
STOP Dropping Block-1;
TURN Herkulex (H-) wheels to left 90° CCW;
// Leftwards motion.
While (BM_1 ~ 3 Not Touch)              // move leftwards until BM_1 ~ 3 touches left wall.
    START Moving leftwards;
    If (Heading > 3°)                    // misalignment detection & correction.
        MOVE H-wheels CW to achieve robot alignment;
    elseif (Heading > -3°)
        MOVE H-wheels CCW to achieve robot alignment;
    endif
endWhile
STOP moving leftwards;
// Rightwards motion.
While (BM_4 ~ 6 Not Touch)              // move rightwards until BM_4~6 touches right wall.
    START Moving rightwards;
    If (Heading > 3°)                    // misalignment detection & correction.
        MOVE H-wheels CCW to achieve robot alignment;
    elseif (Heading > -3°)
        MOVE H-wheels CW to achieve robot alignment;
    endif
endWhile
STOP moving rightwards;
MOVE leftwards for 2 ~ 3 sec.            // this movement is to keep robot little away from right wall
// of stairs, to prevent rubbing with right side-wall, before stepping to next staircase.
Turn H-wheels 90° CW.
LIFT Block-2 upwards (~18cm)            // determined by mechanical lock.
While (ToF3 > 0.5cm)                    // move fwd until ToF3 measures < 0.5cm.
    MOVE FWD;
endWhile
STOP Moving fwd;
While (ToF2_Range_Down < 0.5)           // drop Block_2 until ToF2 senses < 0.5cm range.
    DROP Block-2;
endWhile
STOP Dropping Block-2;
LIFT Block-3 upwards (~18cm);          // determined by mechanical lock.
Go to Step#1;
endif
STOP sTetro;                            // finished climbing up the staircase.

```

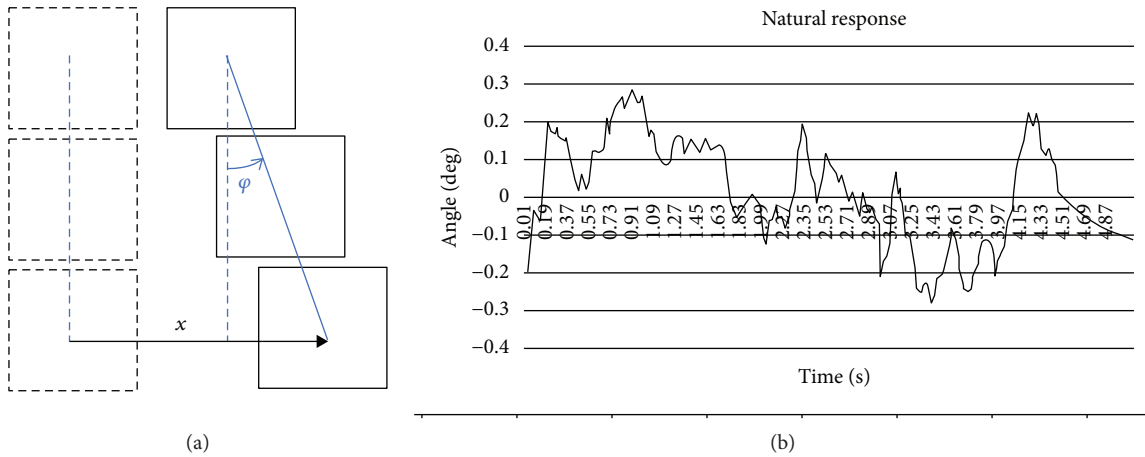


FIGURE 11: (a) Schematic diagram for deviation process. (b) Time plot of angle under natural response.

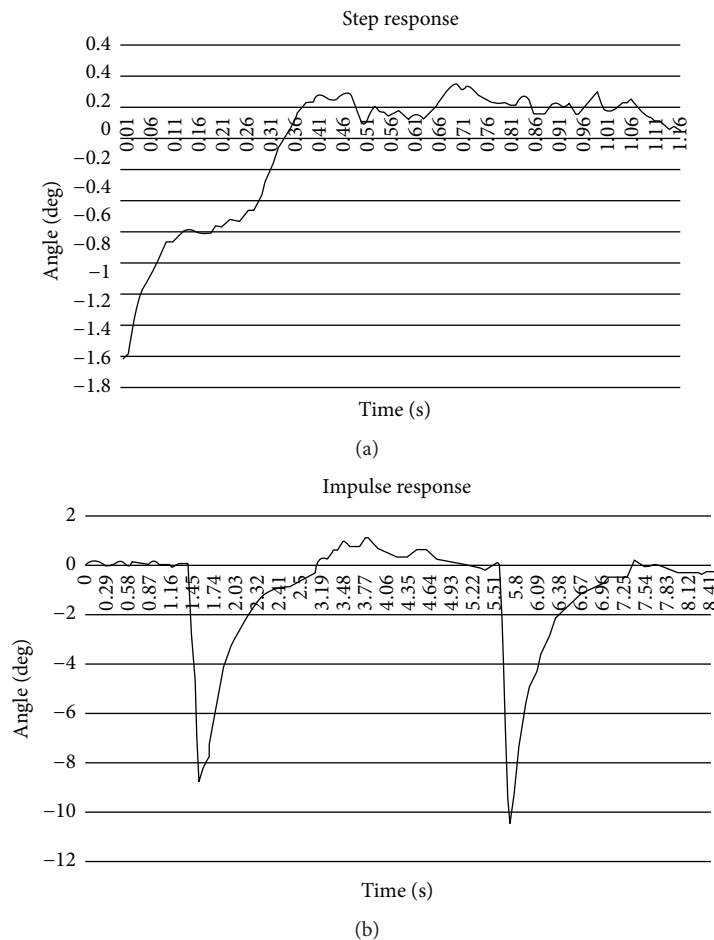


FIGURE 12: (a) Step response. (b) Impulse response of the system.

is simulated by kicking block-1 back to create an abrupt change in the deviation angle. Figure 12(b) shows the time plot of the angle under impulse disturbance.

The plot of the step and impulse responses of this system coincide with the simulation with exceptions to settling time,

peak time, and the peak value. The exceptions are mainly attributed to differences in motor specifications and input conditions between real and simulation settings. However, the trends in both plots are the same indicating that the PID controller is able to regulate the movement under unit

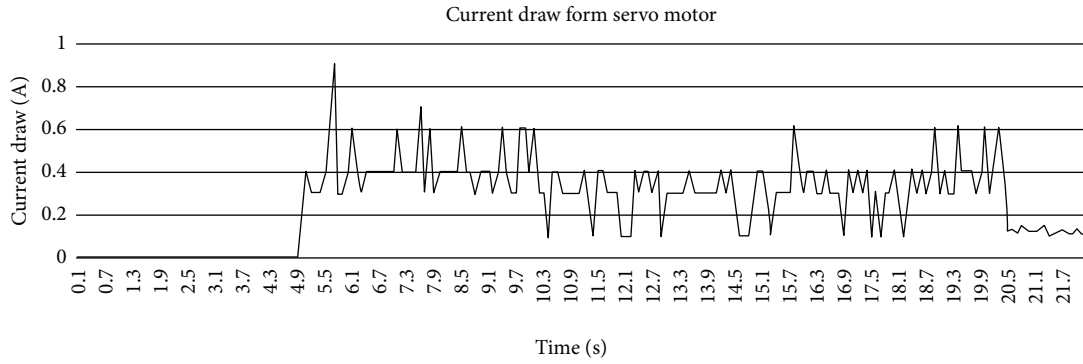


FIGURE 13: Current versus time plot during lifting of block-1.

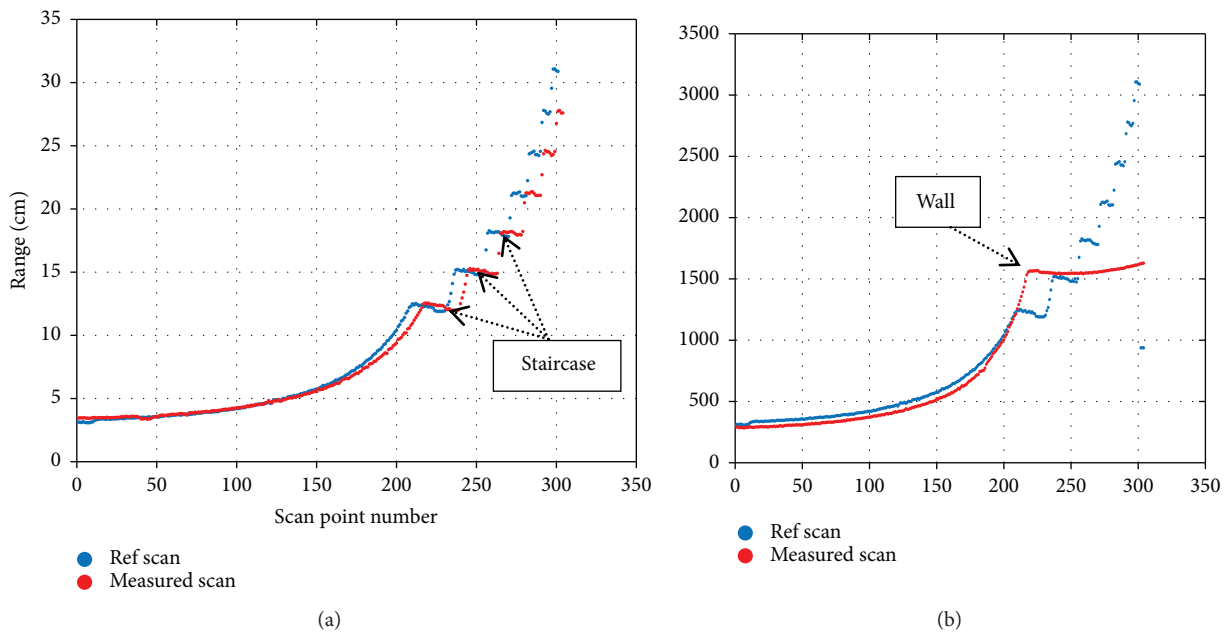


FIGURE 14: (a) Reference and measured scan matching for staircase. (b) Wall detection in measured scan.

step and impulse inputs. The simulation and experiments performed clearly indicate that the gain value obtained from the system modelling for the *sTetro* platform is usable in regulating the sideways motion of the unit.

6.2. Torque Experiment on the Conveyor Motor. This section presents our efforts to experimentally validate the estimated torque for the conveyor motor using the model in Section 4.1 to ensure stable operation. The torque of the servo motor is not measured directly. Instead, it is approximated by the current drawn by the concerned conveyor motor. An ACS714 current sensor is integrated onboard the *sTetro* robot and used to measure the current draw by the motor. Figure 13 shows the current versus time plot during the lifting of block-1.

The figure above presents the change in torque for the concerned servo motor. The values of the current vary around 0.3 A which falls within the acceptable working range

for the servo motor. Based on the manufacturing specification of Herkulex drs-0201 motor used, the torque generated by the servo during this process is approximately 1.13 kgf cm while lifting block-1 which coincides with the theoretical value. This validates the theoretical predictions for the torque needed for the conveyor motor.

6.3. Staircase Detection Experiment. For the smooth transition from the flat surface to the staircase, a staircase recognition algorithm is tested in this work. In Figure 10(a), the experimental setup is shown for the stair recognition test. The *sTetro* robot fixed with the 2D lidar in the vertical position is put in facing stairs, and a reference scan is taken. After that, the robot can be put in any arbitrary orientation and it starts scanning its environment by turning the counterclockwise direction and scan matching algorithm is activated. In Figure 14(a), the range scan matching is given for one scan shot. Staircases are clearly detected when the lidar is nearly

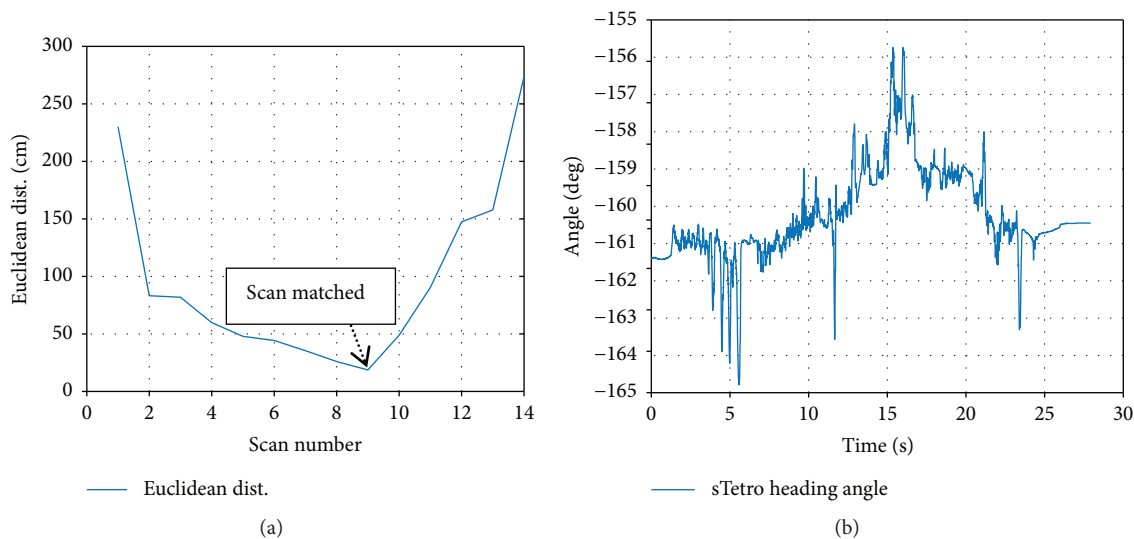


FIGURE 15: (a) Euclidean distance plot for each scan matching outcome. (b) Heading angle of *sTetro*.

facing the stairs. However, if the robot is not facing the stairs, but facing, for example, a wall, then the measured scan gives a one big step in range data, as shown in Figure 14(b), which is clearly distinguishable from the staircase pattern.

In Figure 15, the plot of Euclidean norm versus scan index is given. As the robot starts turning at any arbitrary angular position but approaches the angular position at which the reference scan was taken, the Euclidean norm gives the minimum value. This angular position declared that the robot has recognized stairs and is now facing the staircase. After recognizing the stairs, the *sTetro* starts approaching the staircase to start the cleaning operation.

Figure 15(b) gives the heading angle change during the left and right motion of the *sTetro*. As the robot starts from a heading angle of about -161.4° , the left/right motion affects the heading/misalignment angle within a tolerance of $\pm 5^\circ$. Beyond this limit, the controller activates for corrective measures as described in the autonomy algorithm.

7. Conclusion

In this paper, we presented the design and testing of a novel modular reconfigurable cleaning robot, *sTetro*, that uses vertical conveyor mechanism as a basis. We put forward system models for the estimation of torque for the conveyor motor and centre of gravity analysis towards platform stability. We also describe in detail the design of a PID controller to ensure stable motion of the *sTetro* robot.

The initialization step, which comprises of staircase recognition during operation, is also tested and shows staircase recognition results are very faithful. Moreover, for smooth autonomous operation, the alignment of the robot with the front riser of the staircase is also detected and results are promising to detect and correct misalignment, if any, using single MEMS IMU. To achieve full autonomy, an algorithm is developed which uses inputs from simple, low-cost, low-power sensors and ensures smooth autonomous staircase cleaning operation, by self-reconfiguration and forward/

left/right motion. The integration of contact sensors, ToF sensors, 2D lidar, and an IMU ensures autonomous navigation over a wide range of flat floors and staircases as well, without human intervention. The experiments performed with the real robot demonstrated the efficacy and validity of the developed system models and control approaches.

Future research work on *sTetro* will focus on the following: (1) incorporation of UWB/vision sensor network for localization and autodocking with charging station; (2) integration of a suspension system with a force sensor for adaptation to multiterrain and steadier performance; (3) development of algorithms that ensure optimal path planning outcomes that maximize area coverage while minimizing computational and energy needs; and (4) development of additional features to improve power management issues.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is financially supported by the National Robotics R&D Program Office, Singapore, under the Grant no. RGAST1702, the Singapore University of Technology and Design (SUTD), and the University of Engineering and Technology Lahore (UET), Pakistan (for granting postdoctoral research leave), which are gratefully acknowledged to conduct this research work.

References

- [1] E. Prassler, A. Ritter, C. Schaeffer, and P. Fiorini, "A short history of cleaning robots," *Autonomous Robots*, vol. 9, no. 3, pp. 211–226, 2000.
- [2] Encyclopedia of Management, "Robotics," 2009, September 2016, <https://www.encyclopedia.com>.
- [3] J. Palacin, J. A. Salse, I. Valganon, and X. Clua, "Building a mobile robot for a floor-cleaning operation in domestic environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 5, pp. 1418–1424, 2004.
- [4] N. Tan, N. Rojas, R. Elara Mohan, V. Kee, and R. Sosa, "Nested reconfigurable robots: theory, design, and realization," *International Journal of Advanced Robotic Systems*, vol. 12, no. 7, p. 110, 2015.
- [5] V. Prabhakaran, M. R. Elara, T. Pathmakumar, and S. Nansai, "hTetro: a tetris inspired shape shifting floor cleaning robot," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6105–6112, Singapore, May 2017, IEEE.
- [6] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 718–724, 2004.
- [7] J. S. Oh, Y. H. Choi, J. B. Park, and Y. F. Zheng, "Complete coverage navigation of cleaning robots using triangular-cell-based map," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 3, pp. 718–726, 2004.
- [8] J. Fink, V. Bauwens, F. Kaplan, and P. Dillenbourg, "Living with a vacuum cleaning robot," *International Journal of Social Robotics*, vol. 5, no. 3, pp. 389–408, 2013.
- [9] T. Akinfiyev, M. Armada, and S. Nabulsi, "Climbing cleaning robot for vertical surfaces," *Industrial Robot: An International Journal*, vol. 36, no. 4, pp. 352–357, 2009.
- [10] V. Prabhakaran, M. R. Elara, T. Pathmakumar, and S. Nansai, "Floor cleaning robot with reconfigurable mechanism," *Automation in Construction*, vol. 91, pp. 155–165, 2018.
- [11] L. Zhang, Y. Yang, Y. Gu, X. Sun, X. Yao, and L. Shuai, "A new compact stair-cleaning robot," *Journal of Mechanisms and Robotics*, vol. 8, no. 4, article 045001, 2016.
- [12] S. Rhim, J.-C. Ryu, K.-H. Park, and S.-G. Lee, "Performance evaluation criteria for autonomous cleaning robots," in *2007 International Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, FL, USA, 2007, IEEE.
- [13] S. C. Wong, L. Middleton, B. A. MacDonald, and N. Auckland, "Performance metrics for robot coverage tasks," in *Proceedings of Australasian Conference on Robotics and Automation*, vol. 27, p. 29, Auckland, New Zealand, 2002.
- [14] S. D. Herbert, A. Drenner, and N. Papanikolopoulos, "Loper: a quadruped-hybrid stair climbing robot," in *2008 IEEE International Conference on Robotics and Automation*, pp. 799–804, Pasadena, CA, USA, May 2008, IEEE.
- [15] M. Eich, F. Grimminger, and F. Kirchner, "A versatile stair-climbing robot for search and rescue applications," in *2008 IEEE International Workshop on Safety, Security and Rescue Robotics*, pp. 35–40, Sendai, Japan, October 2008, IEEE.
- [16] J. Liu, Y. Wang, S. Ma, and B. Li, "Analysis of stairs-climbing ability for a tracked reconfigurable modular robot," in *IEEE International Safety, Security and Rescue Robotics, Workshop, 2005*, pp. 36–41, Kobe, Japan, June 2005, IEEE.
- [17] E. Z. Moore, D. Campbell, F. Grimminger, and M. Buehler, "Reliable stair climbing in the simple hexapod 'RHex'," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, pp. 2222–2227, Washington, DC, USA, 2002, IEEE.
- [18] K. Kikuchi, K. Sakaguchi, T. Sudo, N. Bushida, Y. Chiba, and Y. Asai, "A study on a wheel-based stair-climbing robot with a hopping mechanism," *Mechanical Systems and Signal Processing*, vol. 22, no. 6, pp. 1316–1326, 2008.
- [19] H. Zhang, J. Zhang, R. Liu, and G. Zong, "Mechanical design and dynamics of an autonomous climbing robot for elliptical half-shell cleaning," *International Journal of Advanced Robotic Systems*, vol. 4, no. 4, p. 47, 2007.
- [20] R. K. Megalingam, A. Prem, A. H. Nair, A. J. Pillai, and B. S. Nair, "Stair case cleaning robot: design considerations and a case study," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, pp. 760–764, Melmaruvathur, India, April 2016, IEEE.
- [21] H. J. Hoogeboom and W. A. Kusters, "The theory of Tetris," *Newspaper of the Dutch Organization for Theoretical Computer Science*, vol. 9, pp. 14–21, 2005.
- [22] S. A. Bouhamed, I. K. Kallel, and D. S. Masmoudi, "Stair case detection and recognition using ultrasonic signal," in *2013 36th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 672–676, Rome, Italy, 2013.
- [23] D. C. Hernández, T. Kim, and K. H. Jo, "Stairway detection based on single camera by motion stereo," in *Modern Approaches in Applied Intelligence*, K. G. Mehrotra, C. K. Mohan, J. C. Oh, P. K. Varshney, and M. Ali, Eds., vol. 6703 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2011.
- [24] A. Perez-Yus, D. Gutierrez-Gomez, G. Lopez-Nicolas, and J. J. Guerrero, "Stairs detection with odometry-aided traversal from a wearable RGB-D camera," *Computer Vision and Image Understanding*, vol. 154, pp. 192–205, 2017.
- [25] A. Sinha, P. Papadakis, and M. R. Elara, "A staircase detection method for 3D point clouds," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 652–656, Singapore, 2014.
- [26] T. Schwarze and Z. Zhong, "Stair detection and tracking from egocentric stereo vision," in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 2690–2694, Quebec City, QC, Canada, September 2015, IEEE.
- [27] M. Bansal, B. Southall, B. Matei, J. Eledath, and H. Sawhney, "Lidar-based door and stair detection from a mobile robot," in *Unmanned Systems Technology XII*, pp. 7692–7698, Orlando, FL, USA, 2010.
- [28] S. Yuyao, M. R. Elara, M. Kalimuthu, and M. Devarassu, "sTetro: a modular reconfigurable cleaning robot," in *4th IEEE/IFTOMM International Conference on Reconfigurable Mechanisms and Robots (ReMAR)*, Delft, Netherlands, June 2018.