GPS-Free Maintenance of A Free-Space-Optical Link Between Two Autonomous Mobiles

Mahmudur Khan, *Member, IEEE*, Murat Yuksel, *Senior Member, IEEE*, and Garrett Winkelmaier, *Member, IEEE*

Abstract—Free-Space-Optical (FSO) communication has the potential to provide optical-level wireless communication speeds. It can also help solve the wireless capacity problem experienced by the traditional RF-based technologies. Despite its capacity advantages, FSO communication is prone to mobility. Since the FSO transceivers are highly directional, they require establishment and maintenance of line-of-sight (LOS) between each other. We consider two autonomous mobile nodes, each with one FSO transceiver mounted on a movable head capable of rotating 360 degree. We propose a novel scheme that deals with the problem of automatic maintenance of LOS alignment between the two nodes with mechanical steering of the FSO transceivers. We design protocols to maintain an FSO link between the mobiles satisfying a minimum received power or signal-to-noise ratio (SNR). We also present a prototype implementation of such mobile node with FSO transceivers. The effectiveness of the alignment protocol is evaluated by analyzing the results obtained from both simulations and also experiments conducted using the prototype. The results show that, by using such mechanically steerable transceivers and a simple auto-alignment mechanism, it is possible to maintain optical wireless links in a mobile setting with nominal disruption.

Index Terms—Free space optical communication, line of sight, autonomous mobiles, prototype

1 Introduction

Free-space-optical (FSO), a.k.a. optical wireless, communication has recently attracted significant interest from telecommunication research and industry, mainly due to the increasing capacity crunch faced by the radio frequency (RF) wireless technologies [2]. The heavily saturated RF bandwidth is becoming more scarce as cellular capacity has mostly hit its limits. FSO communication (FSOC) has the potential to complement the traditional RF networks. It uses the unlicensed optical spectrum and mostly uses the same basic optoelectronic technology as the fiber optic communications. FSOC can easily reach very high modulation speeds (up to 10 Gbps [3]). Compared to RF, it can provide much higher bandwidth channel to trasnfer large volumes of data. It can also provide connectivity in unfavorable conditions, e.g., presence of RF jamming or interception [4]. A highly useful feature of FSOC is its inherent signal security due to the containment of FSO signals behind walls and their low probability of interception and detection [5], [6]. Also, high directionality of FSOC provides high spatial reuse and larger network capacity.

Applications of the existing FSO communication has been mostly in immobile settings and at high altitudes (e.g., space, satellite, building tops). Fixed FSOC techniques have been studied to remedy small vibrations [7], swaying of the buildings [8], interference and noise [9]. Line-of-sight (LOS) scanning, tracking and alignment have also been studied for years in satellite FSOCs [10], [11], [12]. These works considered the use of mechanical auto-tracking [13], [14] or beam steering [15]. There are also several scenarios involving mobile nodes that can benefit from the various advantages offered by FSOC. Secure command and control of mobile units in combat, sharing of high-resolution imagery and guidance data in next-generation air-traffic control, airborne internet, and rapid communication deployment in disaster recovery are a few examples of these scenarios [16].

FSOC is very useful for signal security and RF challenged environments. So, equipping military robots like PackBots with FSO/VLC (VLC stands for visible light communication) transceivers is a potential application area for FSOC. Using FSOC/VLC with PackBots [17] instead of RF communication can prevent RF interception and jamming from enemies in war zones. Another potential application of FSOC can be equipping robots like the NASA K10 robots [18] with such transceivers for Lunar/Mars exploration. Two K10 robots can communication with each other using FSOC at a lot more faster speed than that using RF communication. Unmanned Aerial Vehicles (UAVs) can also be equipped with such high-speed FSO/VLC transceivers that enables a large set of applications involving transfers of very large wireless data [19]. UAVs can be applied in both military or civil missions which require many sensors and can generate large amounts of data to be transferred to other UAVs or a ground station. Currently UAVs communicate through RF which offers a maximum capacity of around 274 Mbps [20]. The higher data rate required for communication links to transmit more information between UAVs can be provided by equipping them with FSO/VLC transceivers.

- *M. Khan and M. Yuksel are with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816. E-mail: mahmudurk@knights.ucf.edu, Murat.Yuksel@ucf.edu.*
- *G. Winkelmaier is with the Department of Electrical and Biomedical Engineering, University of Nevada, Reno, NV 89557. E-mail: winkg14@gmail.com.*

Manuscript received 7 Dec. 2015; revised 27 July 2016; accepted 11 Aug. 2016. Date of publication 25 Aug. 2016; date of current version 3 May 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2016.2602834

Despite the advantages over RF communication, the major challenge faced by FSOC is its vulnerability against mobility [2], [21]. Mobile FSOC requires effective maintenance of LOS. Since the optical beam is highly focused, the existence of LOS is not enough. The transmitter and the receiver must be aligned; and the alignment must be maintained to compensate for any sway or mobility in the nodes. In this paper, we propose a novel scheme showing the feasibility of maintaining optical wireless communication in a mobile setting with minimal disruption using mechanically steered transceivers and a simple auto-alignment mechanism. We consider two autonomous nodes/robots moving in random directions, each initially unaware of the location of the other. We focus on the case where the mobiles have an FSO transceiver each, mounted on a mechanically steerable head (which could as well be a simple arm) capable of rotating 360 degree. In this paper, we also present a prototype implementation of such mobile FSO nodes. We show that using such mechanical steering capability to control the rotation of the transceivers, the problem of LOS maintenance can be dealt with effectively *without global positioning system (GPS) but merely with an orientation device such as compass*.

Maintaining a directional communication link would be trivial if a location service like GPS is available. However, for RF-challenged or indoor environments, solutions that do not utilize RF signals or GPS are needed. Within this context, our major contributions include:

- a theoretical framework for GPS-free maintenance of FSO links so that calculation of mechanical steering parameters (e.g., angular speed of the rotating arm) is feasible,
- a protocol for maintaining the FSO link at a desired minimum Signal-to-Noise Ratio (SNR) or link quality,
- an approach that uses *only* the optical signals without any usage of RF signals for exchanging information among the mobiles, and
- an approach that does not need multiple transceivers or elements to detect the movement direction of the other mobile.

The rest of the paper is organized as follows: In Section 2, we review the literature for wireless link maintenance between multiple mobiles or in robot teams. Then, in Sections 3, 4 and 5, we describe our proposed method for maintaining communication link between two autonomous mobiles using FSOC. Sections 6 and 7 presents a simulation-based evaluation of our approach. In Section 8 we describe the prototype in detail. In Section 8.1, we provide some initial experimental results. Finally, we conclude in Section 9.

2 Related Work

Maintaining wireless communication links between mobiles has been an attractive problem to both research and industry because of its desirability in many application areas, ranging from robotics to vehicular systems. In [22], FSO communication between two unmanned aerial systems (UASs) hovering in a given location and orientation is considered. The authors developed a novel alignment model and analyzed against simulated and multirotor platforms. In [23], the capabilities of a mechanical gimbal is investigated for use in a ground-to-UAV FSO communications link. In both of these works, laser transmitters are considered and also either one or both nodes are stationary. while our work considers infrared VCSEL/LED transmitters, and communication between two mobile nodes.

A method for establishing a free-space optical-communication link among nearby balloons with the aid of GPS, RF, camera, and communication with a ground station is presented in [24]. In [25], a similar method is proposed that uses predicted movement for maintaining optical-communication lock with nearby ballons. In both [24] and [25], LOS alignment between the communicating nodes is first achieved using GPS information or using a camera to localize the neighbor node. During this phase, RF communication is used. Only after locating the neighbor node, a pointing mechanism is used to align the FSO transceivers of the neighboring nodes. Then optical wireless communication is used only for exchanging data. The optical wireless link is not used for maintaining the link.

A hybrid RF-FSO system is presented in [26], where, the authors developed a system consisting of an MRR(Modulating Retro-Reflector)-FSO link with a tracking optical terminal, a conventional RF link and a deployable pod to provide a relay node bridging the FSO link to the operator and the RF link to the robot. This is a hybrid approach consisting of both RF and FSO. Also the FSO communication is achieved using laser. Our approach consists pure optical wireless communication, and we consider LEDs and VCSELs as transmitters.

In terms of localizing and tracking, [27] presented experimental studies of strategies for maintaining end-to-end communication links for search-and-rescue and surveillance to a base station. The multi-robot team used in the experiment consisted of four unmanned ground vehicles (UGVs) built from radio-controlled scale model trucks each equipped with a laptop computer, odometry, stereo camera, GPS receiver, and a small embedded computer with 802.11b wireless connectivity, called the Junction Box (JBox). Likewise, Parker et al. [28] deployed a team of mobiles to form an indoor sensor network. In their approach the mobile sensors use different techniques such as acoustic sensing, laser scanning and a vision system (such as, a camera) for localization. Shoval et al. [29] measured the relative position and orientation between two mobile robots using a dual binaural ultrasonic sensor system. Each robot was equipped with a sonar transmitter that sends signals to two receivers mounted on the other robot. In [30], a laser-based pedestrian tracking system in outdoors is presented using GPS-enabled mobile robots. In this tracking method, all the robots share the tracking data with each other, so that individual robots always recognize pedestrians that are invisible to other robots.

Our approach considers absence of accoustic or ultrasonic sensing, vision system like camera, laser scanning, radio communication, and a central base station. It assumes FSOC between two mobile nodes, *without any form of GPS*, and *only uses point-to-point distance measurement*. We consider only FSO communication for both establishing/maintaining the link and exchanging information/data. It also assumes autonomy for the mobiles and works in a completely distributed manner.

3 Technical Approach

3.1 Problem Statement and Assumptions

For the problem of maintaining an FSO link between two mobiles, we make the following assumptions that the nodes:

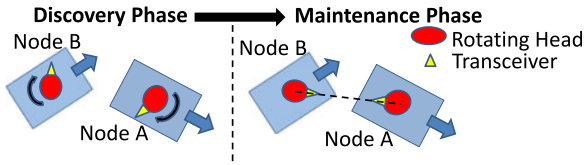


Fig. 1. Establishment of an FSO link between A and B.

- are in a GPS-free environment with no medium of communication available other than the FSO link;
- are mobile and completely autonomous;
- move on both straight lines and curved lines;
- are each equipped with an Internal Measurement Unit (IMU). The IMU consists of an accelerometer, gyroscope and a compass/magnetometer giving them the sense of their own direction and movement; and
- are each equipped with a mechanically steerable head (with which they can scan the entire 360 degree) that is mounted with a full-duplex FSO transceiver.

Our algorithm has two main stages: (i) Detection of line-of-sight and establishment of an FSO link, which we call the discovery phase, and (ii) Maintaining the FSO link. The basic idea for our neighbor discovery approach is to rotate the transceivers of each node with a given angular speed. The nodes start a three way handshake by sending search messages. Upon reception of a search message a node responds with an acknowledgment. The sender of the search message then responds to this acknowledgment completing the three-way handshake. Thus, LOS is detected and a full-duplex FSO link is established. While establishing this link, the nodes exchange the information about their speed and heading direction with each other. Then, in the "maintaining the link" phase, each node uses these speed and heading information to determine the angular velocity at which to rotate its transceiver to maintain the LOS, thus the FSO link with the neighbor node.

3.2 Discovery

Let us consider the case where two mobile nodes are moving in random directions but within the transmission range of each other. The steerable heads of both the nodes rotate continuously in the same direction (both clockwise/ both counter clockwise) but with different angular velocities. While rotating, both the nodes periodically send SYN frames through their respective transceivers. A node stops sending this signal when it receives either a SYN_ACK frame as a reply to its own SYN frame or a SYN frame from the other node. As an answer to the SYN_ACK the sender again sends ACK ensuring detection and establishment of LOS. For simplicity, let us follow the case depicted in Fig. 1, in which node **A** sends a SYN, node **B** replies with SYN_ACK and **A** replies with an ACK. When **A** sends out its first ACK frame, it changes its internal state to ALIGNED with Node **B** and same is true for **B** when it receives the ACK. At this point, **B** and **A** can start exchanging DATA frames as the link is established. The pseudocode of a simple discovery protocol is given in Algorithm 1.

A key issue with LOS discovery among two autonomous nodes is the time it will take for the two nodes to finally get their transceivers in LOS and hence make the above SYN-ACK exchange. If the nodes are rotating their heads in similar speeds, it may take longer time to discover each other. Increasing the rotation speed will not work after a point. A more detailed study of the issue of how fast and which direction the nodes should turn their heads needs to be done. *In this paper, we focus on the next stage of the problem,* i.e., *preserving an existing LOS link between two mobiles.*

Algorithm 1. Discovery: LOS Detection and Establishment

```
DiscoverNeighbor()
 1: {Initialize the state variables}
 2: State = SEARCH //Aligned or Search state?
 3: Done = FALSE //Full duplex or not
 4: {Continuously rotate head and send search signal (SYN)}
 5: while State = SEARCH do
 6:       Send SYN
 7:       if SYN is received then
 8:             State = ALIGNED
 9:             Send SYN_ACK
10:       end if
11:       if SYN_ACK is received then
12:             State = ALIGNED
13:             Done = TRUE
14:             Send ACK
15:       end if
16: end while
17: {Wait for SYN_ACK if not received}
18: while Done = FALSE do
19:       Send SYN
20:       if SYN_ACK is received then
21:             Done = TRUE
22:             Send ACK
23:    end if
24: end while
```

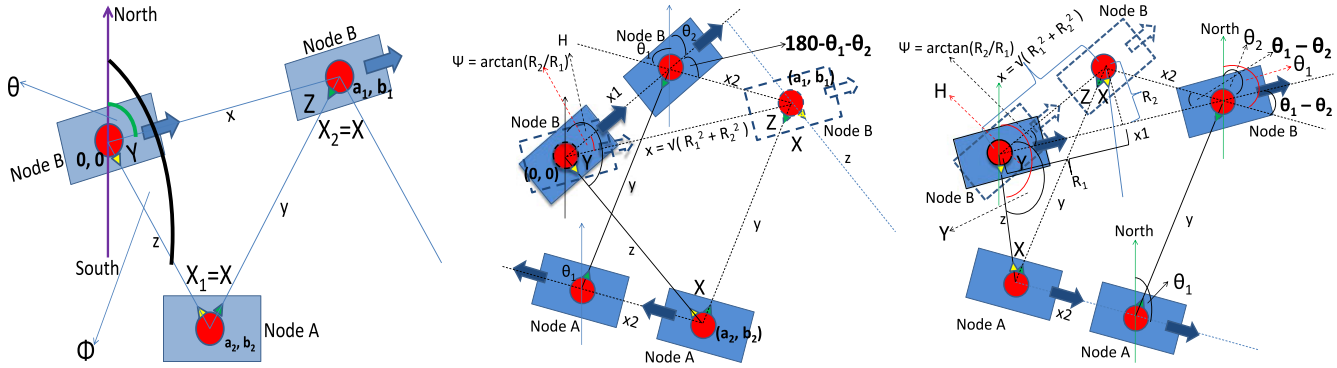
3.3 Maintaining the Link

In this stage, we assume that an FSO link has been established between the two mobiles, but the goal is to maintain this link. While establishing the link, a node conveys the information about its velocity, the direction in which it is moving and the orientation of its head to the other node. This information is used by the other node to set the angular velocity and the direction of rotation (clockwise/counterclockwise) of its head to maintain the link.

After establishing the FSO link, the nodes maintain the link with the aid of mechanical steering. Using the determined angular velocity and finding which way to rotate (clockwise/counterclockwise), the head is steered accordingly to maintain the FSO link. If one or both of the nodes change velocity or direction or both, then they need to recalculate their heads' angular velocity and direction of rotation, which we detail in the next two sections. The pseudocode of a simplistic link maintenance protocol is given in Algorithm 2.

4 Autonomous Maintenance of an FSO Link

Assuming that two nodes A and B have already detected the LOS and established an FSO link between them, there are several calculations and decisions to be made in order to maintain the FSO link between each other. These include which *direction to rotate* their heads, calculation of the proper *angular*



(a) Case I (b) Case II (c) Case III

Fig. 2. Determining angle of rotation.

velocity of the rotation, and calculation of the *distance between the nodes*. Last but not the least, the nodes should know how frequent and which specific information to exchange among each other so that the calculations and decisions could be made in a timely manner. We address these issues below.

Algorithm 2. Maintenance: Tune Angular Velocity of the Head

{Global variables}
Dist_Nodes //*Distance between the two mobiles in meters*
R_{max} //*Maximum communication range of the transceiver in meters*
{Head rotation variables}
AngularVelocity //*Degrees per second*
AngularDirection //*CW or CCW*

{My movement/mobility variables}
Velocity //*My speed and direction of movement*
Orientation //*Head's orientation*

{Neighbor's movement/mobility variables}
NVelocity //*Neighbor's speed and direction of movement*
NOrientation //*Head's orientation*

{Periodically exchange velocity, direction & head orientation}
{Called every t_x seconds}
ExchangeInfo()
1: Send $< Velocity, Orientation >$
2: Receive $< NVelocity, NOrientation >$
{Periodically check the link and re-tune the angular velocity of the head}
{Protocol A: Called every t_x seconds}
Maintain($\alpha_{\max}$)
1: $\alpha =< Angle\ of\ Deviation > /\theta$
2: {Check if the link is still up}
3: **if** $Dist\_Nodes > R_{max}$ **then**
4: DiscoverNeighbor()
5: **else**
6: {Check if the link has deviated more than $\alpha_{\max}$}
7: **if** $\alpha > \alpha_{\max}$ **then**
8: Recalculate *AngularVelocity*
9: Recalculate *AngularDirection*
10: **end if**
11: **end if**

4.1 The Angular Velocity: Three Cases

There are three cases according to the relative directions and positions of the nodes. We detail these cases for autonomously calculating the angular velocity of the nodes' heads so that the link can be maintained.

Case I: One Moving, One Stationary. As depicted in Fig. 2a, let's consider the case when A is stationary and B is mobile. B continues to move in the direction it was already moving. Let us assume that B moves a distance of x in some given time t (very small amount of time in tens of milliseconds). To sustain the link, both A and B will need to rotate their heads. If the angle of rotation for the head of A is X_1 and for the head of B is X_2. From Fig. 2a it can be seen that

$$\angle X_1 + \angle Y + \angle Z = 180^\circ, \tag{1}$$

and that the path traversed by B is a straight line

$$\angle Z + \angle X_2 + \angle Y = 180^\circ. \tag{2}$$

(1) and (2) show that $\angle X_1 = \angle X_2$. Let this angle of rotation be X, where $\angle X = \angle X_1 = \angle X_2$. Then, the angle of rotation $\angle X$ can be determined using the following formula:

$$\frac{x}{\sin X} = \frac{y}{\sin Y} = \frac{z}{\sin Z}, \tag{3}$$

where x, y and z are the edges of the triangle in Fig. 2a. We can, then, calculate X by:

$$X = \arcsin \frac{x \sin Y}{y}. \tag{4}$$

Since we know B's velocity, $\vec{v_b}$, from the discovery phase, we can calculate x by measuring the time difference between the two positions of B and applying

$$Distance = Velocity \times Time. \tag{5}$$

That is, $x = v_b \times t$. Next, we can also calculate $\angle Y$ as it is the angle between the direction of B's motion and the orientation of its head. To find y, we look at Fig. 2a. Let us consider the position of B at the moment of LOS discovery as the origin $(0, 0)$ of the reference frame. After time t, B's position is (a_1, b_1) and let A's position be (a_2, b_2). Then, we can write $a_1 = x \sin\theta$, $b_1 = x \cos\theta$, $a_2 = z \sin\phi$, and $b_2 = z \cos\phi$, where θ is the angle between the compass and the direction of B's motion, ϕ is the angle between the compass and the orientation of the head. We can now calculate $y = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$, which means everything needed to calculate X in (3) is complete. Finally, the angular velocity

for A's head will be $\angle X/t$, since we know that

$$Angular\ Velocity = \frac{Angular\ Displacement}{Time}. \quad (6)$$

We will describe how to find z later in Section 4.2.

Case II: One Node Eastbound, One Node Westbound. Fig. 2b portrays another case where A is going westbound and B eastbound. Let θ_1 be the angle between the compass axis and the direction of motion of the first node, A and θ_2 be the angle between the compass axis and the direction of motion of the second node, B. Then, eastbound represents $\theta_2 = [0^\circ, 179^\circ]$ and westbound represents $\theta_1 = [180^\circ, 359^\circ]$. Assume that B moves x_1 and A moves x_2 after discovering LOS between each other. Similar to Case I, we can assume A as stationary and B as moving with relative velocity $\vec{v_a} + \vec{v_b}$ which gives B's relative displacement, x, as $\vec{x} = \vec{x_1} + (-\vec{x_2})$. Here, $x = \sqrt{R_1^2 + R_2^2}$ and $\psi = \arctan(R_2/R_1)$, where $R_1 = x_1 + x_2 \cos(180^\circ - \theta_1 - \theta_2)$, $R_2 = x_2 \sin(180^\circ - \theta_1 - \theta_2)$, and ψ is the angle between the original direction and the relative direction of B's motion. Similar to Case I, considering B's position at the moment of LOS discovery as the origin $(0,0)$ of the global reference frame, we can calculate A's position (a_2, b_2) at the moment of LOS detection and B's apparent position (a_1, b_1) at distance x. This would give us $y = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$. And $\angle Y = \angle H - \theta_2 - \psi$. Here, $\angle H$ represents the orientation of the head of Node B with respect to the compass axis. Finally, we then again use (3), (4), (5) and (6) to determine the angular velocity, $\angle X/t$.

Case III: Both Nodes Eastbound or Westbound. The last case is portrayed in Fig. 2c, where A and B are both going eastbound, with velocities $\vec{v_a}$ and $\vec{v_b}$, respectively. Again, similarly to Case II, we can assume A as stationary and B as moving with velocity $\vec{v_b} - \vec{v_a}$. Here $\vec{x} = \vec{x_1} + (-\vec{x_2})$, where $x = \sqrt{R_1^2 + R_2^2}$ and $\psi = \arctan(R_2/R_1)$. Here, $R_1 = x_1 - x_2 \cos(\theta_1 - \theta_2)$ and $R_2 = x_2 \sin(\theta_1 - \theta_2)$. Here, $\theta_1 =$ angle between the compass axis and the direction of motion of Node A. And θ_2 = angle between the compass axis and the direction of motion of Node B. Similar calculations as in Cases I and II give us y. We then calculate $\angle Y = \angle H - \theta_2 + \psi$. Finally, we apply (3), (4) and (5) and (6) to find the angular velocity, $\angle X/t$.

These calculations for determining the angular velocity are applicable for nodes moving on both straight lines and curves. We assume the curve to be a concatenation of a large number of very small straight lines. The distance $x = v_b X t$ would be very small since we assume t to be very small ($10\ ms - 1s$). Although it is possible to further improve this calculation for curvature movements, we leave that for future work.

4.2 Determination of z

We represent the distance between the two mobile nodes at the moment they discover the LOS as z. This distance (z) is measured only at the start of the maintenance phase. To find z, we assume availability of an optical distance measurement device, which are available in three categories: interferometry, time-of-flight (TOF) and triangulation methods [31]. The FSO transceivers can use the TOF technique to measure z. TOF refers to the time it takes for a pulse of energy to travel from its transmitter to an observed object and then back to the receiver. If light is used as energy source, the relevant parameter involved in range counting is the speed of light, i.e., roughly 30 cm/ns. A TOF system measures the round trip time (RTT) between a light pulse emission and the return of the pulse echo resulting from its reflectance off an object. When LOS is established between the two nodes, this technique is applicable.

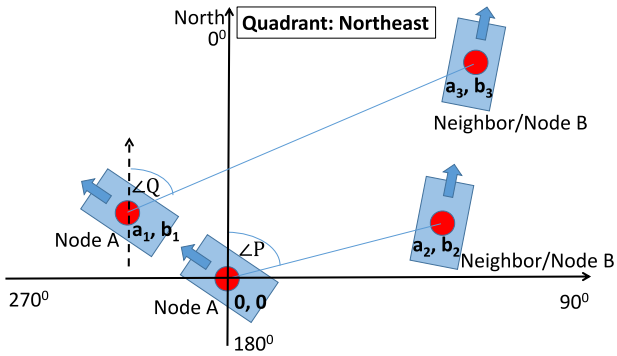


Fig. 3. Rotation direction: Neighbor in northeast quadrant.

Another method may be to measure the time between sending a "Hello" packet to the neighbor and receiving a response for that packet. For this case, while calculating the RTT, we also need to consider the processing time of the receiving node (μs to ns) and transmission time of (μs) the control packet. We assume that processing time of the receiver is constant and known to the sender or the processing time can be included in the response packet. The transmission time or time to insert the control packet into the propagation medium can be calculated by dividing the size of the control message by the line rate (upto 155 Mbps [2] for VCSELS and LEDs). Since RTT is representative of traveling twice the distance and must therefore be halved to find the actual range to the target [31]. In this case, RTT is

$$RTT = t_{recv} - t_{send} - t_{proc} - t_{trans}, \quad (7)$$

where, t_{recv} is the time when response is received from the receiver (Node B), t_{send} is the time when initial signal was sent by sender (Node A), t_{proc} is the processing time of the nodes and t_{trans} is the time to insert the control packet into the propagation medium.

4.3 Rotation: Clockwise (CW) or Counterclockwise (CCW)?

Another important decision to make for the nodes is which way to rotate: clockwise or counterclockwise. Considering two nodes A and B again, the decision depends on two parameters: orientation of the heads and the direction of motion of the nodes. Fig. 3 depicts an example. Assume that, Node A was at location $(0,0)$ and Node B was at location (a_2, b_2) at the start of the "Maintaining The Link" phase. After a given time, Node A moves to (a_1, b_1) and Node B Moves to (a_3, b_3). Here $\angle P$ is the orientation of the head of node A with respect to the compass axis at the start of the "Maintaining The Link" phase. $\angle Q$ is the orientation of the head of node A with respect to the compass axis after a given time. If $\angle P < \angle Q$, then rotation should be Counterclockwise.

TABLE 1
Neighbor in Northern Quadrants

Condition	*Rotation*
$\angle P < \angle Q$	CW
$\angle P > \angle Q$	CCW
$\angle P = \angle Q$	None

If $\angle P > \angle Q$, then rotation should be Clockwise and if $\angle P = \angle Q$, there is no need for the nodes to rotate their heads. Tables 1 and 2 show all the possible cases which aids in deciding whether the rotation should be clockwise or counterclockwise.

5 Exchange Protocols to Maintain the Link

A crucial part of establishing the nodes' autonomy is to let the nodes decide how fast to turn their heads autonomously. We assume that the nodes will exchange their velocity and signal quality values periodically over the FSO link itself. These periodic exchanges will allow the nodes to recalculate how fast they should turn their heads so that the FSO link stays up. We considered two different simplistic protocols for recalculating the angular velocity of the rotating head.

5.1 Protocol A: Maximum Angle of Deviation

Every time the nodes exchange their information, the receiver node will see if it has deviated greater than a preset threshold. If so, the receiver node will recalculate the angular speed of its head. We assume that the nodes can turn their heads as fast as needed, which is realistic since we only consider walking speeds.

Let t_x be the time period of information exchanges between the two nodes. Further let α be the ratio between the Angle of Deviation (θ_d) of the receiver from the the normal of the other node's beam and the Divergence Angle (θ), and $\alpha_{\max}$ be the maximum allowed α before recalculation of rotational speed is performed. At every t_x, α is checked, and if $\alpha > \alpha_{\max}$ then the angle of rotation (X) for the mobile FSO nodes are recalculated using (4), (5), (6).

During simulations, it was observed that $\theta_d >> X$ when $\alpha > \alpha_{\max}$. So instead of updating X by directly adding/subtracting θ_d, it is updated as follows:

$$\angle X = \angle X\left(1 \pm \frac{\theta_d}{\theta}\right). \tag{8}$$

Also, a value of $\alpha > 1$ means that the link is down. Algorithm 2 lays down one particular implementation of Protocol A as a pseudocode.

5.2 Protocol B: Minimum SNR

In this protocol, every time the nodes exchange their information, the receiver node will see if its received Signal-to-Noise Ratio is less than a preset threshold. If so, the receiver node will recalculate the angular speed of its head. This design is particularly useful if quality of the FSO link is important.

Let γ be the difference (in dB) between the received SNR of the receiver and the receiver's minimum required SNR, and $\gamma_{\min}$ be the minimum allowed γ before recalculation of rotational speed is performed. At every t_x, γ is checked, and if $\gamma < \gamma_{\min}$ then the angle of rotation (X) for the mobile FSO nodes are recalculated. A value of $\gamma < 0$ means that the link is down. The angle of rotation is updated using (8).

TABLE 2
Neighbor in Southern Quadrants

Condition	*Rotation*
$\angle P < \angle Q$	CCW
$\angle P > \angle Q$	CW
$\angle P = \angle Q$	None

TABLE 3
Mathematical Notations

Symbol	Meaning
θ	Angle of Divergence (degrees)
α	Angle of Deviation / θ
α_{max}	Maximum allowed α
γ	Received SNR - Minimum required SNR (dB)
γ_{min}	Minimum allowed γ (dB)
t_x	Time period of information exchange(s)
N_{rec}	No. of Recalculations
R_{max}	Maximum transmission range of transceiver (m)
ζ	Radius of receiver (cm)
P_t	Transmitter's source power (dBm)
S	Receiver's sensitivity (dBm)

6 Simulation Setup

To gain insight into effectiveness of our approach, we performed simulations using MATLAB [32]. We considered walking speeds and reasonably capable robots for our nodes, e.g., Packbots [17]. We concentrated on the "Maintaining the Link" phase, and assumed that the nodes had discovered each other. We detail our simulation setup and assumptions, followed by results below. Table 3 lists the meanings of each mathematical symbol we used in our models.

6.1 Transceiver Coverage Model

A key part of the simulation is to model transmission and field-of-view areas of an FSO transceiver, which follow the Lambertian law. To ease the computations, we approximated an FSO transceiver's coverage area L as the combination of a triangle and a half circle, which was shown to approximate the Lambertian coverage with a negligible error in [33]. Fig. 4 illustrates the key parameters: R, the height of the triangle; θ, the divergence angle and R_{max}, the maximum reachable range. The radius of the half circle is $R\tan\theta$, and R can be found by $R_{max} = R + R\tan\theta$. Then, the coverage area of the transceiver, L, can be derived as

$$L = R^2\tan\theta + \frac{1}{2}\pi(R\tan\theta)^2. \tag{9}$$

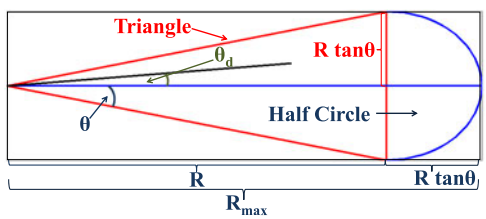


Fig. 4. Coverage area of an FSO transceiver as "Triangle + Half Circle".

The received power of an FSO transceiver is subject to Lambertian loss due to radial distance from the axis of propagation (see Fig. 4), atmospheric attenuation and geometric attenuation [33]. The maximum communication range of an FSO transceiver is affected by these losses. The maximum range R_{max} that can be reached by an FSO transceiver (or the maximum reachable range) is dependent on the transmitter's source power P dBm, the receiver's sensitivity S dBm, the radius of the transmitter β cm, the radius of the receiver (on the other receiving FSO node) ζ cm, the divergence angle of transmitter θ mRad, the visibility V km, and the optical signal wavelength λ nm. The maximum solution of the following inequality gives us R_{max}

$$-(P_t + S) < 10\log_{10} e^{-\sigma R} + 10\log_{10}\left(\frac{\zeta}{\gamma + 200R\theta}\right)^2. \quad (10)$$

6.2 Divergence Angle

During our simulations, we tried different divergence angles for the transceivers. We assumed that, for a transceiver, both the transmitter's divergence angle and the receiver's field-of-view are the same. Lasers operate with 0.5 mRad (0.0286 degree) to 2.5 mRad (0.1432 degree) divergence angles, while VCSELs with 2.5 mRad (0.1432 degree) to 75 mRad (4.2972 degree) and LEDs with 60 mRad (3.4377 degree) to 200 mRad (11.4592 degree) [33]. For our simulations, we considered VCSELs and LEDs as transmitters and considered 3, 5, 7.5, and 10 degree as divergence angles.

6.3 Minimum SNR in Protocol B

Protocol B aims to keep the quality of the FSO link above a threshold. The performance of a receiver degrades as a result of several factors, including line-width, relative intensity noise (RIN) of the source, and receiver noise. These effects have an impact on the maximum transmission distance and signal coverage area. The performance of an optical digital link is measured by the bit error rate (BER). Conversely, in an analog optical link, the performance of the receiver is measured by SNR. An example of such an analog system is the optical wireless CATV system [34], where multiple analog or digital TV signals (or both) are combined by means of subcarrier multiplexing (SCM) into a single analog signal, which is then transmitted over an optical link. To provide good picture quality, the analog signal must have an SNR much greater than typical (e.g., 14 to 17 dB) for non-return-to-zero (NRZ) signal. For analog TV channels, the National Association of Broadcasters (NAB) recommends an SNR $>$ 46 dB. For a digital TV channel with QAM-256 modulation and forward error correction (FEC), a typical SNR $>$ 30 dB is required [34], [35]. For our simulation purposes, we chose the minimum required SNR of 30 dB for Protocol B.

6.4 Node Size and Configuration

Our main simulation scenario is to evaluate two Packbots exploring a region of interest while maintaining an FSO link between each other. So, we considered the nodes as Packbots having length and width of 75 cm and 40 cm respectively [17]. We considered a fixed value of R_{max} (100 meters) for different values of the divergence angle. We considered receiver's sensitivity of -43 dBm, visibility of 20 km, optical signal wavelength of 1550 nm, transmitter radius of 0.3 cm and receiver radius of 3.75 cm. The required source power for the transmitter was calculated from (10).

6.5 Sensor Reading Errors

We consider both nodes to be equipped with an Internal Measurement Unit (IMU) each. An IMU consists of an accelerometer, gyroscope and a magnetometer. These sensors are used by the nodes to measure their own speed and heading direction that they share with each other. These sensors are sensitive and contains measurement errors like noise, a bias, scale factor error, g-sensitivity, cross-axis sensitivity etc. Disturbances in the magnetic field near the magnetometer can induce errors. This disturbances include power lines, motors, residual magnetism in the vehicle's chassis etc. [36]. We considered the presence of these errors in our simulations and associated different error values to the measurement of initial distance [31], speed [37] and heading [38] of the nodes. We denoted the different error values as follows:

- *NE:* No error was considered.
- *EC1:* Error values between -0.25 m/s and 0.25 m/s were randomly chosen from a truncated Gaussian distribution and added to the speed measurement.
- *EC2:* Error values between -0.50 m/s and 0.50 m/s were added to the speed measurement.
- *EC3:* Error values between -0.75 m/s and 0.75 m/s were added to the speed measurement.

Moreover, for *EC1*, *EC2* and *EC3*, error values between -3° and 3° were added to the heading measurements and error values between -0.25 m and 0.25 m were added to the initial distance measurements (determination of z) between the nodes.

7 Simulation Results

We evaluated the performance of our approach using the two exchange Protocols A and B for recalculating the angle of rotation autonomously. We compare the percentage of time the FSO link was down during our simulations. We evaluated Protocols A and B under various ranges for the parameters shown in Table 3. We performed simulations for different values of the divergence angle θ (2.5, 5, 7.5 and 10 degree), $\alpha_{\max}$ (0.25, 0.5 and 0.75), $\gamma_{\min}$ (5 dB, 6 dB, and 7 dBb), and t_x (10ms to 1s). We first considered Packbots moving on straight lines with speeds between 0m/s to 2.5 m/s (or 9 km/h) [17]. We randomly chose the initial positions of the nodes. We also randomly picked their initial speed and heading direction for each simulation run. For each simulation run, after every two seconds, the speed of the nodes were increased by 10 percent of its current speed for the first ten seconds. For the next ten seconds, the speed of the nodes were reduced by 10 percent of its current speed. If the link duration reached twenty seconds or if the nodes were out of each others' communication range the simulation was ended. For calculating the percentage of time the link was down for each simulation run, we checked every 10 ms if a receiver node was within the coverage area of the transmitting node, i.e., for protocol A we checked if $\alpha > 1$ and for protocol B we checked if $\gamma < 0$. The values $\alpha > 1$ and $\gamma < 0$ were counted as misses. Dividing the count of misses by the link duration value gave us the percentage of link down value. For each parameter

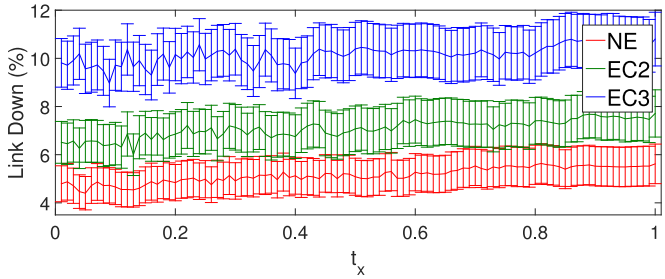


Fig. 5. Effect of t_x and sensor errors on link maintenance for $\alpha_{max} = 0.25$ and $\theta = 7.5$ degree (with 70 percent confidence interval).

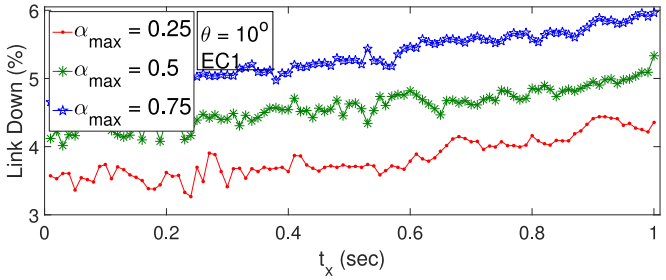


Fig. 7. Effect of α_{max} on link maintenance for $\theta = 10^0$.

combination, we calculated the average percentage of link down time and the number of message exchanges over 400 simulation runs.

7.1 Using Protocol A

We simulated the protocol for different values of $\alpha_{\max}$ (0.25, 0.5 and 0.75), and t_x (10ms to 1s). In this section, we report the results where angle of deviation is used as the metric for deciding when to re-tune the angular speed of the head's rotation (i.e., Protocol A). We simulated with two mobility scenarios: (i) both nodes moving on straight lines and (ii) one node on a curve and the other on a straight line.

7.1.1 Both Nodes on Straight Lines

Fig. 5 depicts the performance of the algorithm for maintaining the FSO link for different values of t_x. The percentage link down time values are plotted along with 70 percent confidence interval. We can see that, as t_x increases the link is down for longer percentage of time. Smaller t_x means more frequent exchange of speed and heading information between the nodes. This helps to update the angular velocity with more accuracy and provides better link maintenance. For example, with error EC2 (explained in Section 6.5), link down time is 6.57 percent for $t_x = 30$ ms and 7.71 percent for $t_x = 100$ ms. We can also observe from Fig. 5 that, the link is down for longer percentage of time with increase in errors in the sensor readings. As we can see, when no sensor error is considered the link is down for ≈ 5 percent of the time. When error is $EC2$, link is down for ≈ 7 percent and when error is $EC3$, link down time is ≈ 10 percent.

Fig. 6 depicts the performance of the algorithm for maintaining the FSO link for different divergence angles when $\alpha_{\max} = 0.25$ and error is $EC1$. The percentage link down time values are plotted along with 95 percent confidence interval. We can observe that the lower the divergence angle, the larger percentage of time the link is down. For example, when $t_x = 50$ ms and $\theta = 2.5$ degree, link down time is 19.33 percent. On the other hand, when $t_x = 50$ ms and $\theta = 10$ degree, link down time is only 3.34 percent. A larger θ with a fixed $R_{\max}(100\text{ m})$ provides a larger coverage area than a smaller θ. So, the larger the divergence angle, the better is the performance.

For a fixed divergence angle, $\theta = 10°$, Fig. 7 shows the link maintenance performance for different values of $\alpha_{\max}$ and sensor error $EC1$. We observe that smaller $\alpha_{\max}$ results in better link maintenance. We can see that, the percentage of time the link is down, is larger for $\alpha_{\max} = 0.25$ than that for $\alpha_{\max} = 0.75$. This is expected since a lower $\alpha_{\max}$ value means that the recalculation of rotational speed is done for less deviation from the sender node's area of coverage. Thus, lower $\alpha_{\max}$ yields more accurate calculation of angle of rotation and thus the link stays up for a longer period of time.

A relevant issue is how the links' duration relates to the link down time. Fig. 8 shows the averages of the percentage of link down time across link duration. The link duration is the length of time the two nodes are both in communication ranges of each other. If the link duration reached 20s simulation was stopped. We can observe that the percentage of down time is larger for the links that had longer duration, e.g., 0 percent down time for links ongoing up to 16s while $0-4$ percent for links ongoing longer than 16s. This is a motivating result since the largest percentage of link down time is only 4 percent for links longer than 16s, and the link down time stays under control.

Fig. 9 displays the relationship between divergence angle and required transmitter power. The average percentage link down time values are plotted along with 95 percent confidence interval. We can observe that for a fixed value of R_{max}, although the performance of the algorithm improves with larger divergence angles, this is achieved at the cost of higher transmitter source power (calculated from [33]). To maintain

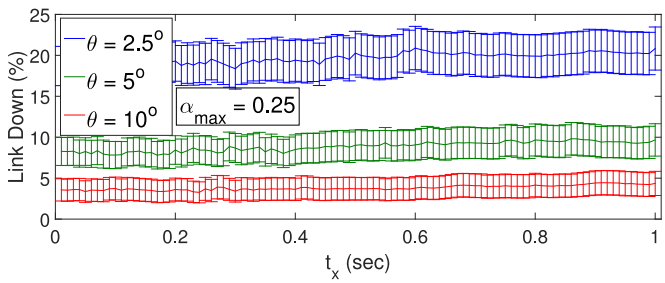


Fig. 6. Effect of divergence angle on link maintenance for $\alpha_{max} = 0.25$ (with 95 percent confidence interval).

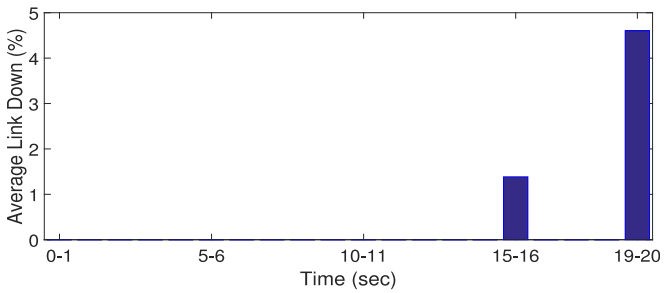


Fig. 8. Distribution of down time percentage across link duration ($\alpha_{\max} = 0.25, \theta = 10^o, t_x = 50$ ms).

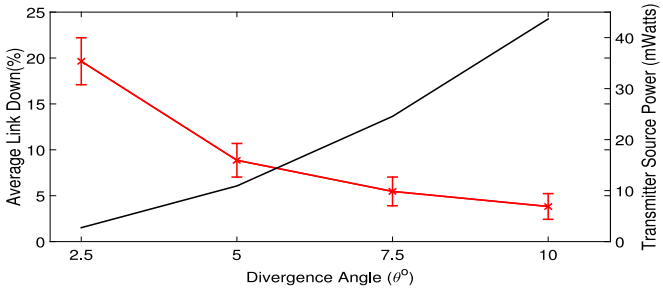


Fig. 9. Divergence angle (θ) versus transmitter source power.

a particular R_{max}, the required transmitter source power increases (Black line) with increase in the divergence angle.

Fig. 10, shows the combined effect of divergence angle and $\alpha_{\max}$ on link down time. We can observe that, for small sensor errors (NE or $EC1$), combination of larger divergence angles and smaller $\alpha_{\max}$ provides better link maintenance. But this behavior is not consistent when sensor errors are high ($EC2$ and $EC3$). For example, with error $EC2$ and $\theta = 2.5$, $\alpha_{\max} = 0.75$ results in smaller link down time than $\alpha_{\max} = 0.25$, which is contrary to what we observed for NE and $EC1$. But for larger divergence angles ($5, 7.5$ and 10 degree), the combination of large θ and small α works well again. This irregularity is caused by the sensor reading errors.

Lastly, Fig. 11 shows the communication and computation overheads, i.e., the number of recalculations (N_{rec}) of rotational speed for different values of t_x and $\alpha_{\max}$. It can be seen that the overhead reduces with increase in t_x. For example, for $\alpha_{\max} = 0.25$ and $\theta = 5$ degree, we empirically find that the overhead decays with a power of -0.96, i.e., $N_{rec} = c * t_x^{-0.96}$ (c is a constant). This is a very promising result since it is possible to attain significant reductions in overhead by making small increases to t_x. In a more

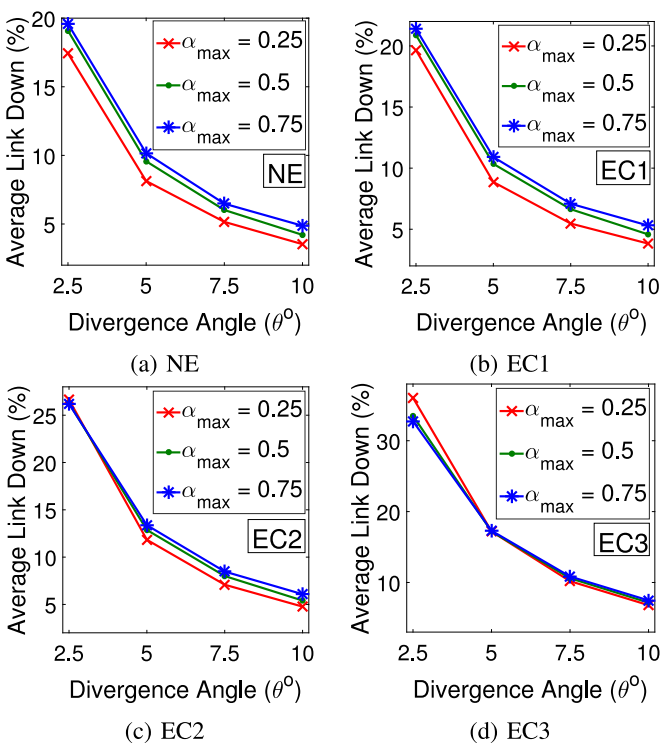


Fig. 10. Combined effect of θ and $\alpha_{\max}$ on link maintenance.

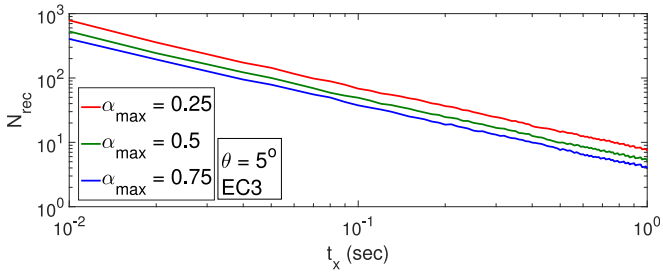


Fig. 11. Overhead for different α_{max} and θ (axes in log scale).

complicated protocol, t_x can be dynamically tuned to leverage this relationship and reduce the recalculation overhead depending on the mobility of the nodes and their relative positions. We can also observe that the overhead decreases as $\alpha_{\max}$ increases. The overhead is higher for $\alpha_{\max} = 0.25$ than that for $\alpha_{\max} = 0.75$.

We have observed that, the proposed scheme provides better link maintenance for smaller values of $\alpha_{\max}$ and t_x, but at the cost of larger overhead. And it also provides better performance when divergence angles are larger but at the cost of larger transmitter source power.

7.1.2 *One Node on a Curve, the Other on a Straight Line*

The angular speed calculations of our approach in Section 7.1.1 assumes that the nodes move on straight lines. One natural question to answer is how the approach would perform if the nodes move on curves. For this case, we considered one node moving on a curve and the other moving on a straight line. We modeled the curved path using the equation of a parabola

$$y = ax^2 + bx + c. \tag{11}$$

We assumed that the node on a straight line is starting from the center of the simulated area, and randomly picked the starting point and the direction of the other node which follows the curve according to (11).

Fig. 12 shows the performance of the link maintenance protocol when one of the nodes moves on a curve. The solid lines show the link down time for the scenario where both nodes move on straight lines. The dotted lines show link down time for the scenario where one node moves on a curve. It can be seen that, the link is down for longer percentage of time when both nodes are not moving on a straight line. But the difference is subtle. Additionally, we observe that, link down time increases with increase in t_x and with larger sensor errors.

Again, Fig. 13, shows the combined effect of divergence angle and $\alpha_{\max}$ on link down time when one node is moving

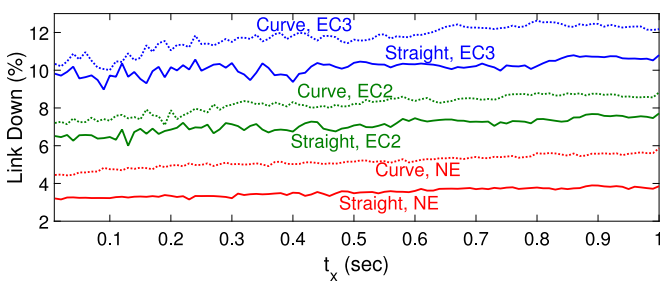
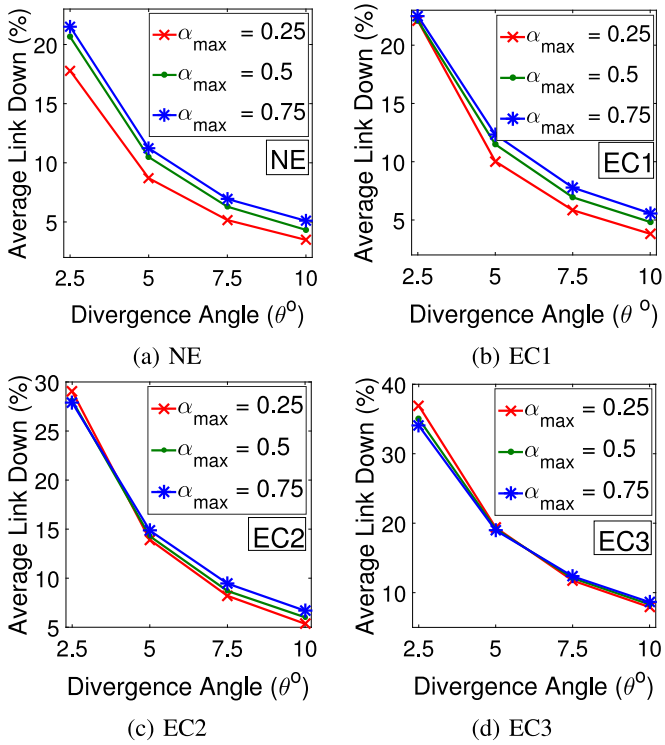


Fig. 12. Effect of t_x and sensor errors on link maintenance.



(a) NE (b) EC1 (c) EC2 (d) EC3

Fig. 13. Combined effect of θ and α_{max} on link maintenance.

on a curve. We again observe that, for no sensor errors (NE), combination of larger divergence angles and smaller $\alpha_{\max}$ provides better link maintenance. But this behavior is not consistent when sensor errors are present ($EC1$, $EC2$ and $EC3$). The combination of large θ and small α works well for larger divergence angles ($\theta > 2.5$ degree for EC1, $\theta > 5$ degree for EC2 and $\theta > 7.5$ degree for EC2).

7.2 Using Protocol B

In this section, we report the results where SNR is used as the metric for re-tuning (i.e., Protocol B). We again considered with two mobility scenarios: (i) both nodes moving on straight lines and (ii) one node on a curve and the other on a straight line. We simulated this protocol also for different values of the divergence angle, $\gamma_{\min}$ (5dB, 6dB and 7dBb), and t_x.

7.2.1 *Both Nodes on Straight Lines*

Fig. 14 depicts the performance of the algorithm for maintaining the FSO link for different values of t_x using Protocol B. Again, the percentage link down time values are plotted along with 70 percent confidence interval. We can see that,

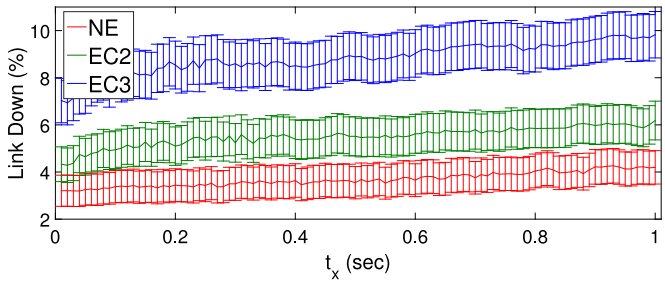


Fig. 14. Effect of t_x and sensor errors on link maintenance for $\gamma_{\min} = 7$ dB and $\theta = 10^\circ$ (with 70 percent confidence interval).

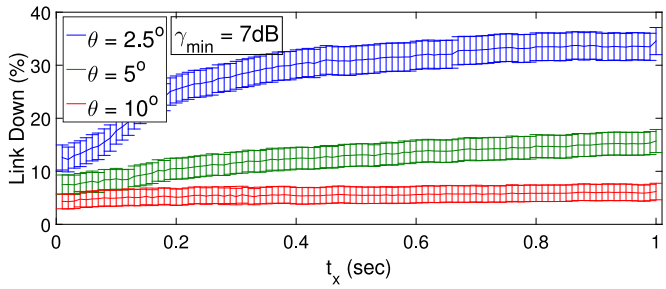


Fig. 15. Effect of θ on link maintenance for $\gamma_{\min} = 7$ dB (with 95 percent confidence interval).

similar to the case using Protocol A, increase in t_x causes the link to be down for longer percentage of time. As discussed earlier, smaller t_x helps to update the angular velocity with more accuracy and provides better link maintenance. We can also observe from Fig. 14 that, the link is down for longer percentage of time with increase in errors in the sensor readings. As we can see, the link down time is smaller when there is no sensor error, compared to when sensor error is $EC2$ or $EC3$.

In Fig. 15, the effect of divergence angle of the transceivers on link maintenance is shown when Protocol B is used for re-tuning the angular velocity of the node's heads. Once again, we observe that, link maintenance is better performed with larger divergence angles.

For a fixed divergence angle, $\theta = 10$ degree, Fig. 16 shows the link maintenance performance for varying $\gamma_{\min}$. We observe that increasing $\gamma_{\min}$ improves the performance. This is also an expected result because for a higher value of $\gamma_{\min}$, the recalculation of angular speed is done for higher difference between the receiver SNR and the threshold SNR. Higher $\gamma_{\min}$ means more accurate calculation of angle of rotation angle and thus better maintenance of the FSO link is achieved.

Fig. 17 displays the combined effect of θ and $\gamma_{\min}$, averaged over all possible values of t_x. It can be seen that increasing the divergence angle yields better link maintenance. Also, increasing the $\gamma_{\min}$ improves the performance. As we can observe, combination of a high divergence angle ($\theta = 10^o$) and a high $\gamma_{\min} = 7$ dB, with error EC1, provides better (≈ 3 percent down time) link maintenance than that (≈ 17 percent down time) obtained using a combination of ($\theta = 2.5$ degree) and $\gamma_{\min} = 5$ dB. Unlike the cases in protocol A, increase in sensor error does not affect this large θ–high $\gamma_{\min}$ combination.

Finally, Fig. 18 shows the communication and computation overhead (i.e., the number of recalculations of rotational speed). Although higher $\gamma_{\min}$ and smaller t_x provides

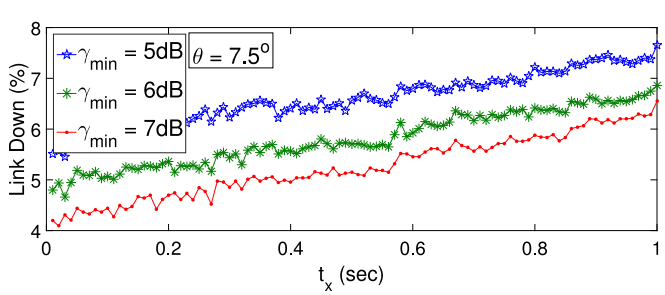


Fig. 16. Effect of $\gamma_{\min}$ on link maintenance for $\theta = 10$ degree.

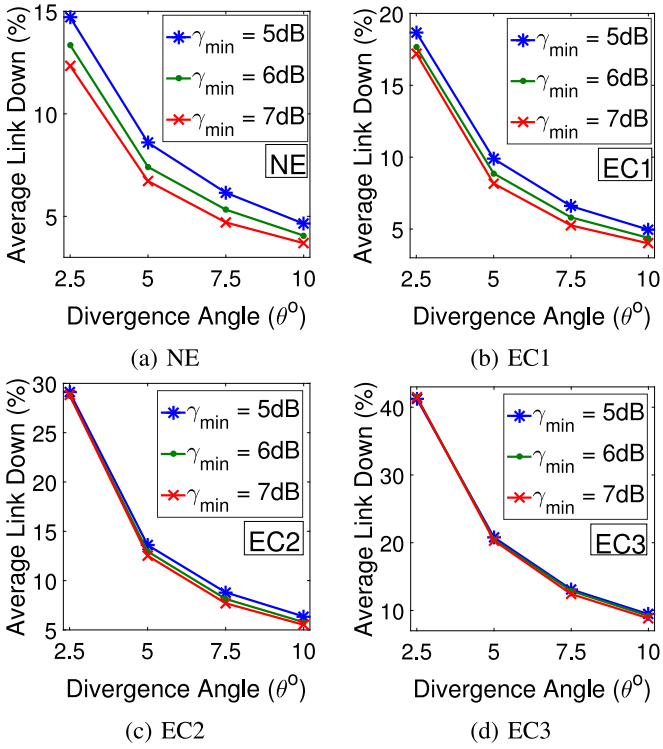


Fig. 17. Combined effect of θ and γ_{min} on link maintenance.

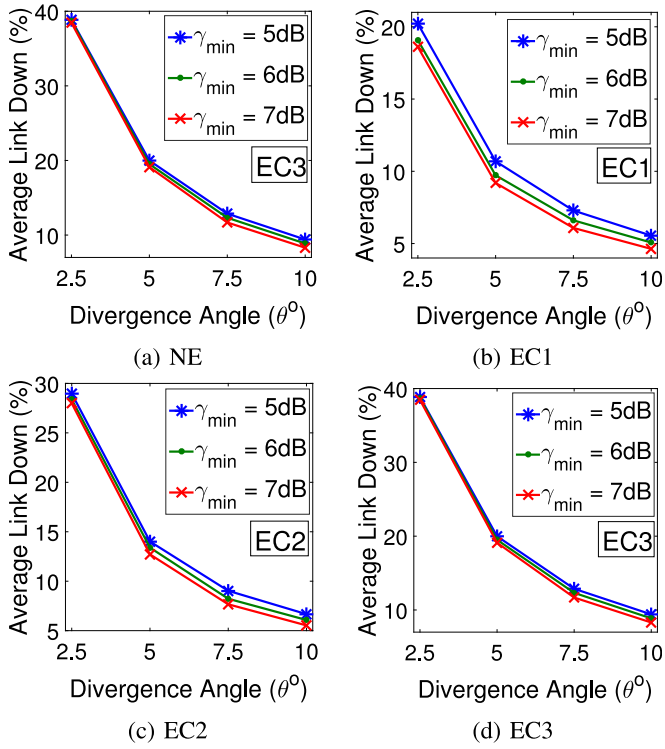


Fig. 20. Combined effect of θ and γ_{min} on link maintenance.

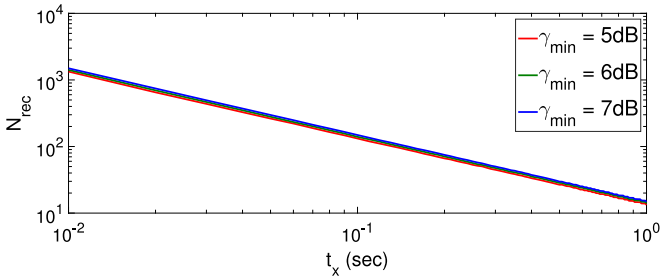


Fig. 18. Overhead for different γ_{min} and θ (axes in log scale).

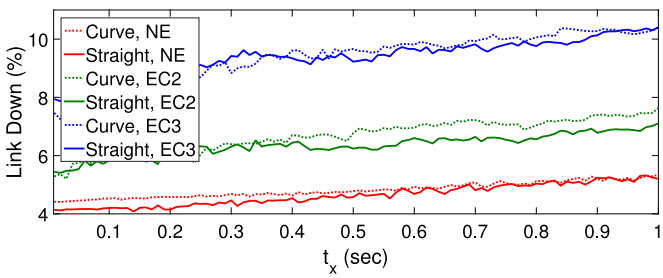


Fig. 19. Effect of t_x and sensor errors on link maintenance.

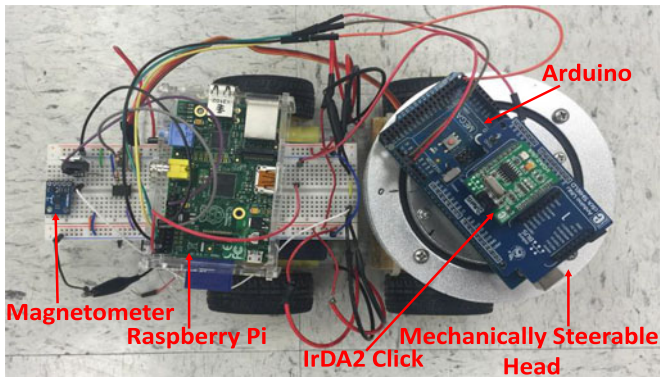


Fig. 21. Bird's eye view of the prototype.

better link performance (Fig. 14), this is achieved at the cost of higher overhead. Again we see that overhead reduces with with increase in t_x. So, it is possible to attain significant reductions in overhead by making small increases to t_x.

7.2.2 *One Node on a Curve, the Other on a Straight Line*

We performed simulations using Protocol B also for the scenario where one node moves on a curve and the other moves on a straight line. We again observe in Fig. 19 that, the link down time is slightly higher when one node is moving on a curve compared to when both nodes are moving on straight lines. Similar to the simulation results in Sections 7.1.1, 7.1.2, and 7.2.1, we see that the FSO link is better maintained for smaller sensor errors. Also, the link maintenance is performed better with smaller t_x (Fig. 19), larger divergence angle, and higher $\gamma_{\min}$ (Fig. 20).

8 Proof-of-Concept Prototype

We designed and built a prototype of the mobile node with a mechanically steerable FSO transceiver by employing commercially available off-the-shelf electronic components. The prototype and a block diagram of it are shown in Figs. 21 and 22, respectively. The main parts of the prototype are: a robot car, a mechanically steerable head, a magnetometer, and IR transceiver. All of these parts are controlled by a Raspberry Pi [39] using separate threads: head control, car control, compass readings, and transmit or receive data. Due to UART compatibility issues of Raspberry Pi, we added an Arduino [40] to handle the

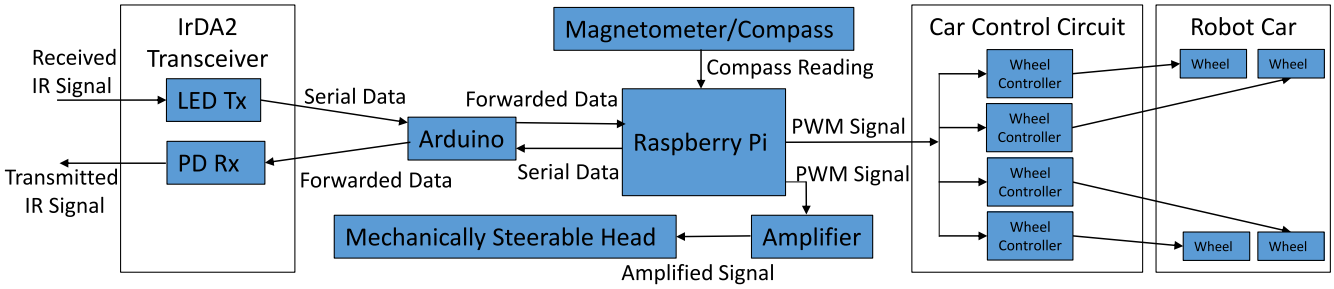


Fig. 22. Prototype block diagram.

transmission and reception of the serial data (via UART) to/from the IR transceiver.

We used the Emgreat 4-wheel Robot Smart Car Chassis Kits car [41] as the mobile node and the Aluminum Robot Turntable Swivel Base [42] as the steerable head on which we mount the IR transceiver. We used IrDA2 Click [43] as the transceiver. It supports IrDA speeds up to 115.2Kbit/s. Integrated within the transceiver module are a photo pin diode, an infrared emitter (IRED), and a low-power control IC to provide a total front-end solution in a single package. This device covers the full IrDA range of 3m using the internal intensity control. The IRED has peak emission wavelength of 900 nm and the angle of half intensity is $\pm 24^o$. And as the magnetometer we used the GY-273 HMC5883L 3-Axis Magnetic Electronic Compass [38].

8.1 Experimental Results

We performed some initial experiments using the prototype to gain insight about the effectiveness of our proposed FSO link maintenance protocol. We considered one node stationary and the other mobile[1]. Also, we considered half-duplex communication mode since the available IrDA2 click transceivers do not work in full-duplex mode. The mobile node worked as a transmitter and the stationary node was the receiver. For data transmission, we used the maximum baud rate of 115.2 Kbps of the IrDA2 click. We considered two types of packets: control packets and data packets. The control packet contained the speed and direction information of the mobile node. The data packet was randomly generated. The packet structures are shown in Fig. 23.

We performed two experiments on the setup shown in Fig. 24a. For both experiments, we kept Node A stationary and Node B moving. We varied the message exchange interval t_x (i.e., the time interval between sending the control packets) and Node B's speed v_x. We measured the throughput of the FSO link as the performance metric.

8.1.1 Experiment I: Angular Change

The goal of this experiment is to show that the protocol can effectively maintain the FSO link while there is an angular change in the relative position of the two nodes. In this experiment, Node B moved on a straight line parallel to the x-axis as shown in Fig. 24a. As Node B moves, both nodes have to tune their heads' angular speeds in order to maintain the link. Node B was stopped after it traveled 8 feet (or 2.44 m), half of it towards the y-axis and the other half away from it. This parallel movement requires changing the heads' angular speeds continuously.

We performed the experiment for $t_x = [1, 2750]$ ms and $v_x = [0.375, 0.75]$ m/s. Fig. 25a shows how the throughput (in Kbps) behaves as t_x varies. We can observe that the throughput was ≈68 Kbps for tx of up to 2,500 ms at speeds 0.375–0.75 m/s. The throughput drops significantly for larger t_x. We can also observe that the maximum value of t_x to achieve a throughput of ≈68 Kbps reduces with increase in speed. As expected, this means that higher speeds require more frequent (smaller t_x) information exchange between the nodes to maintain the same level of throughput. For example, for $v_x = 0.375$ m/s, the throughput was ≈68 Kbps for 1 ms $\leq t_x \leq 2,500$ ms. Likewise, for $v_x = 0.50$ m/s, the throughput was ≈68 Kbps for 1 ms $\leq t_x \leq$1600 ms.

Fig. 25b displays the effect of $\alpha_{\max}$ on the maximum value of t_x to maintain a throughput of ≈68 Kbps. For our experiments, we used $\alpha_{\max}$ of 0.125 and 0.25. We can observe that, for both cases, maximum t_x reduces with increase in velocity. We can also see that the maximum t_x is larger for $\alpha_{\max} = 0.25$ than that for $\alpha_{\max} = 0.125$. We also performed experiments with $\alpha_{\max} = 0.375$, but for this case even with $t_x = 1$ ms, the throughput was ≈50 Kbps.

8.1.2 Experiment II: No Angular Change

The goal of this experiment is to show the sanity of the proposed alignment protocol. In this scenario also, we kept Node A stationary on the x-axis and Node B mobile but there was no angular change in the relative positions of the nodes. First, we performed experiments with Node B moving along the x-axis towards Node A (top Fig. 24b). For all values of t_x and v_x, both nodes correctly calculated the head rotation angles to be 0 degree and successfully maintained the FSO link. The throughput remained ≈68 Kbps for all the trials. Then, we performed experiments with Node B moving on the x-axis away from Node A as shown in Fig. 24b (bottom). In this case also, all the packets sent by Node B was received by Node A correctly.

Control Packet Identifier	Packet Number	Car Speed	Car Heading	Data
1 byte	2 bytes	1 byte	1 byte	3 bytes

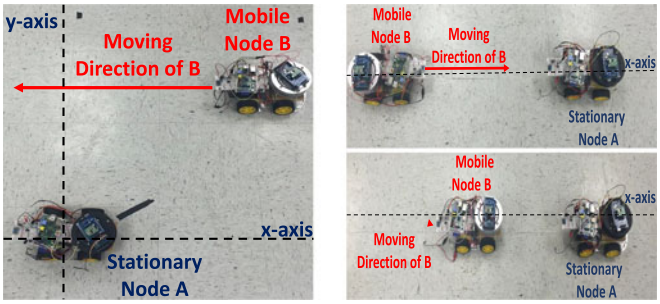
Control Packet

Data Packet Identifier	Data
1 byte	7 bytes

Data Packet

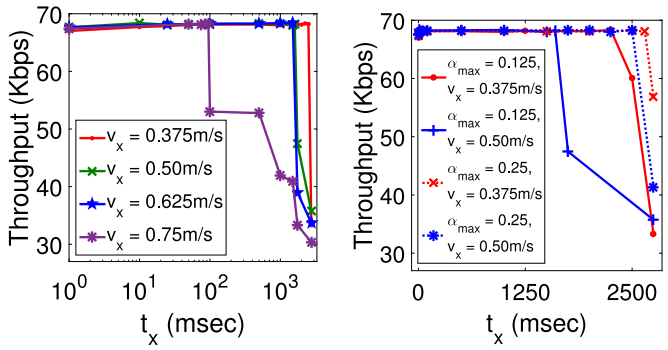
Fig. 23. Packet structure.

1. Videos explaining the proposed method and sample experiments are provided here: https://drive.google.com/folderview?id=0B3IGA4_FJqpXTURPR1BDdS1XTDA&usp=sharing



(a) Scenario I (b) Scenario II

Fig. 24. Experiment scenarios.



(a) Effect of t_x (b) Effect of α_{max}

Fig. 25. Effect of t_x and α_{max} on throughput for different car speed.

9 Summary and Future Work

We presented a novel approach to overcome the problem of maintaining an FSO link between two autonomous mobile nodes. Each of the nodes is equipped with a mechanically steerable head (or arm) on which an FSO transceiver is mounted. Using the proposed algorithm to control the mechanically steered head, the FSO transceivers on the nodes can maintain a communication link successfully. We outlined all possible cases for calculating the angular velocity of the nodes' heads and the direction of the heads' rotation so as to maintain the FSO link. We also presented two different protocols for deciding when to recalculate the angular velocity. One is based on the deviation of the receiver node from the transmitter's coverage area, the other based on a minimum received SNR. We also presented a prototype implementation of the above mentioned mobile FSO nodes. For evaluating our proposed algorithm, we performed MATLAB simulations and real experiments using the developed prototype. We showed through both simulation and experimental results that, using a simple protocol to control the mechanically steerable head, the FSO transceivers on the nodes can maintain a communication link successfully.

We assumed the height of the FSO transceivers to be same, i.e., we considered the nodes traveling at the same height. For nodes positioned at different heights or non-flat terrains, heads capable of rotating both in horizontal and vertical planes can be considered [19]. Nodes with faster traveling speeds can also maintain FSO links using such transceivers as long as they are in each others' communication range for adequate duration of time. Also, we considered FSOC between two mobile nodes. More future work is needed to explore the scenarios involving more than two such nodes. Interference from multiple neighbor nodes would be an interesting problem to tackle in this case.

In our prototype, the current communication speed is limited by the IrDA2 Click (maximum 115.2 Kbps). This limits the speed of the data transfer between the nodes regardless of the IR transceivers capabilities. Also, the maximum range of the IR transceiver is limited to three meters. Further, in the experiments, we considered one stationary and one mobile node with a half-duplex transceiver on each. To reap the full potential of the proposed FSO link maintenance protocol, full-duplex transceivers are required. As future work, we plan to develop full-duplex optical transceivers for improving the prototype. The prototype can be further improved by combining the circuits for the robot car, the mechanical head and the transceiver in a single PCB board. Another possible line of future work is to explore multi-transceiver designs for the mobile nodes. It is possible to equip each node with multiple transmitters and/or receivers and use them for detecting the movement direction of the other node [2]. Such redundancy will help both during the discovery and maintenance phases of our approach.

Acknowledgements

This work was supported in part by NSF awards 1321069 and 1422354, ARO DURIP W911NF-14-1-0531, and NASA Space Grant NNX10A. During most of this work, the authors were with the University of Nevada, Reno, NV 89557. An earlier version of this work appeared in IEEE WCNC 2014 [1].

References

[1] M. Khan and M. Yuksel, "Maintaining a free-space-optical communication link between two autonomous mobiles," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2014, pp. 3154–3159.
[2] A. Sevincer, M. Bilgi, and M. Yuksel, "Automatic realignment with electronic steering of free-space-optical transceivers in MANETs: A proof-of-concept prototype," *Ad Hoc Netw.*, vol. 11, no. 1, pp. 585–595, Jan. 2013.
[3] MRV optical communication systems, 2011. [Online]. Available: http://www.mrv.com/
[4] E. Leitgeb, J. Bregenzer, M. Gebhart, P. Fasser, and A. Merdonig, "Free space opticsbroadband wireless supplement to fiber networks, in free-space laser communication technologies XV," *Proc. SPIE*, vol. 4975, pp. 57–68, 2003.
[5] D. Zhou, P. G. LoPresti, and H. H. Refai, "Enlargement of beam coverage in fso mobile network," *J. Lightw. Technol.*, vol. 29, pp. 1583–1589, 2011.
[6] S. Trisno, T.-H. Ho, S. D. Milner, and C. C. Davis, "Theoretical and experimental characterization of omnidirectional optical links for free space optical communications," in *Proc. IEEE Military Commun. Conf.*, 2004, vol. 3, pp. 1151–1157.
[7] S. Arnon and N. S. Kopeika, "Performance limitations of free-space optical communication satellite networks due to vibrations-analog case," *SPIE Opt. Eng.*, vol. 36, no. 1, pp. 175–182, Jan. 1997.
[8] S. Arnon, "Effects of atmospheric turbulence and building sway on optical wireless-communication systems," *OSA Opt. Lett.*, vol. 28, no. 2, pp. 129–131, Jan. 2003.
[9] A. J. C. Moreira, R. T. Valadas, and A. M. O. Duarte, "Optical interference produced by artificial light," *ACM/Springer Wireless Netw.*, vol. 3, pp. 131–140, 1997.
[10] R. Gagliardi and S. KARP, "Optical Communications," New York, NY, USA: Wiley, 1976.
[11] S. G. Lambert and W. L. Casey, *Laser Communications in Space*. Norwood, MA, USA: Artech House, 1995.
[12] S. Arnon and N. Kopeika, "Laser satellite communication network-vibration effect and possible solutions," *Proc. IEEE*, vol. 85, no. 10, pp. 1646–1661, 1997.

[13] E. Bisaillon, et al., "Free-space optical link with spatial redundancy for misalignment tolerance," *IEEE Photonics Technol. Lett.*, vol. 14, no. 2, pp. 242–244, Feb. 2002.
[14] M. Naruse, S. Yamamoto, and M. Ishikawa, "Real-time active alignment demonstration for free-space optical interconnections," *IEEE Photonics Technol. Lett.*, vol. 13, no. 11, pp. 1257–1259, Nov. 2001.
[15] Y. E. Yenice and B. G. Evans, "Adaptive beam-size control scheme for ground-to-satellite optical communications," *SPIE Opt. Eng.*, vol. 38, no. 11, pp. 1889–1895, Nov. 1999.
[16] D. Zhou, P. G. LoPresti, and H. H. Refai, "Evaluation of fiber-bundle based transmitter configurations with alignment control algorithm for mobile FSO nodes," *J. Lightw. Tech.*, vol. 31, pp. 249–356, 2013.
[17] iRobot. (2015). [Online]. Available: http://www.irobot.com/us/learn/defense/packbot
[18] Nasa k10 robots. (2015). [Online]. Available: http://www.nasa.gov/centers/ames/K10/
[19] M. Khan and M. Yuksel, "Autonomous alignment of free-space-optical links between UAVs," in *Proc. 2nd Int. Workshop Hot Topics Wireless.*, 2015, pp. 36–40.
[20] D. Weatherington and U. Deputy, "Unmanned aircraft systems roadmap, 2005–2030," *Deputy, UAV Planning Task Force, OUSD (AT&L)*, 2005.
[21] A. Ashok, M. Gruteser, N. Mandayam, J. Silva, M. Varga, and K. Dana, "Challenge: Mobile optical networks through visual mimo," in *Proc. MobiCom*, 2010, pp. 105–112.
[22] A. Kaadan, D. Zhou, H. H. Refai, and P. G. LoPresti, "Modeling of aerial-to-aerial short-distance free space optical links," in *Proc. IEEE Integr. Commun. Navigation Surveillance Conf.*, 2013, pp. 1–12.
[23] A. Harris, J. J. Sluss, H. H. Refai, and P. G. LoPresti, "Alignment and tracking of a free-space optical communications link to a UAV," in *Proc. 24th Digital Avionics Syst. Conf.*, 2005, pp. 1–C.
[24] R. W. DeVaul, E. Teller, C. L. Biffle, and J. Weaver, "Establishing optical-communication lock with nearby balloon," US Patent App. 13/346,645, Jan. 9, 2012.
[25] R. DeVaul, E. Teller, C. Biffle, and J. Weaver, "Using predicted movement to maintain optical-communication lock with nearby balloon," US Patent App. 14/108,542, Dec. 17, 2013.
[26] J. L. Murphy, M. S. Ferraro, W. S. Rabinovich, P. G. Goetz, M. R. Suite, and S. H. Uecke, "Control of a small robot using a hybrid optical modulating retro-reflector/RF link," *Proc. SPIE*, vol. 9080, 2014, Art. no. -90 801F–12. [Online]. Available: http://dx.doi.org/10.1117/12.2052942
[27] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Network connectivity and performance in robot teams," *J. Field Robot.*, vol. 25, pp. 111–131, 2008.
[28] L. E. Parker, B. Kannan, X. Fu, and Y. Tang, "Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 2488–2493.
[29] S. Shoval and J. Borenstein, "Measuring the relative position and orientation between two mobile robots with binaural sonar," in Presented at the *ANS 9th Int. Topical Meet. Robot. Remote Syst.*, Seattle, Washington, 2001.
[30] M. Ozaki, M. Hashimoto, T. Yokoyama, and K. Takahashi, "Laser-based pedestrian tracking in outdoor environments by multiple robots," in *Proc. 37th Annu. Conf. IEEE Ind. Electron. Soc.*, 2011, pp. 197–202.
[31] M.-C. Amann, T. Bosch, M. Lescure, R. Myllyla, and M. Rioux, "Laser ranging: A critical review of usual techniques for distance measurement," *Opt. Eng.*, vol. 40, no. 1, pp. 10–19, 2001.
[32] Matlab. (2015). [Online]. Available: http://www.mathworks.com/products/matlab/
[33] M. Yuksel, J. Akella, S. Kalyanaraman, and P. Dutta, "Free-space-optical mobile ad hoc networks: Auto-configurable building blocks," *Wireless Netw.*, vol. 15, no. 3, pp. 295–312, Apr. 2009.
[34] R. Ramirez-Iniguez, S. M. Idus, and Z. Sun, *Optical Wireless Communications: IR for Wireless Connectivity*. New York, NY, USA: CRC Press, 2008.
[35] W. Ciciora, J. Farmer, and D. Large, *Modern Television Technology–Video, Voice and Data Communications*. San Francisco, CA, USA: Morgan Kaufmann, 1999.
[36] E. Abbott and D. Powell, "Land-vehicle navigation using GPS," *Proc. IEEE*, vol. 87, no. 1, pp. 145–162, Jan. 1999.
[37] H. H. Liu and G. K. Pang, "Accelerometer for mobile robot positioning," *IEEE Trans. Ind. Appl.*, vol. 37, no. 3, pp. 812–819, Jun. 2001.
[38] 3-axis digital compass ic hmc58831. (2015). [Online]. Available: https://www.parallax.com/downloads/hmc5883l-3-axis-digital-compass-ic-datash
[39] Raspberry pi. (2015). [Online]. Available: https://www.arduino.cc/en/reference/servo
[40] Ieik uno R3 board atmega328p. (2015). [Online]. Available: http://www.amazon.com/IEIK-Board-ATmega328P-Cable-Arduino/
[41] Emgreat 4-wheel robot smart car. (2015). [Online]. Available: http://www.amazon.com/Emgreat-4-wheel-Chassis-Encoder-Arduino/
[42] Aluminium robot turntable swivel base. (2015). [Online]. Available: http://www.ebay.com/itm/Aluminium-Robot-Turntable-Swivel-Base-2-DOF-PTZ-2pcs
[43] Irda2 click. (2015). [Online]. Available: http://www.mikroe.com/click/irda2/

Mahmudur Khan received the BSc degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, in 2011 and the MS degree in computer science and engineering from the University of Nevada at Reno (UNR), in 2015. He is currently working towards the PhD degree in the ECE Department at the University of Central Florida. His research interests include the area of free-space-optical communications, wireless communications, and UAV communications. He is a member of the IEEE.

Murat Yuksel received the BS degree in computer engineering at Ege University, Izmir, Turkey, in 1996, and the MS and PhD degrees in computer science from RPI in 1999 and 2002, respectively. He is an associate professor in the ECE Department at the University of Central Florida (UCF), Orlando, FL. Prior to UCF, he was with the CSE Department at the University of Nevada—Reno (UNR), Reno, NV, as a faculty member until 2016. He was with the ECSE Department at the Rensselaer Polytechnic Institute (RPI), Troy, NY, as a postdoctoral associate and a member of adjunct faculty until 2006. He worked as a software engineer at Pepperdata, Sunnyvale, CA and a visiting researcher at AT&T Labs and Los Alamos National Lab. His research interests are in the area of networked, wireless, and computer systems with a recent focus on big-data networking, UAV networks, optical wireless, public safety communications, device-to-device protocols, economics of cyber-security and cybersharing, routing economics, network management, and network architectures. He has been on the editorial board of *Computer Networks*, and published more than 100 papers in peer-reviewed journals and conferences and is a co-recipient of the IEEE LANMAN 2008 Best Paper Award. He is a senior member of the IEEE, and a senior and life member of the ACM.

Garrett Winkelmaier received the BS degree in electrical engineering from the University of Nevada, Reno, in 2016. His research interests include unmanned autonomous systems and mathematics. He is currently working with AboveNV as an engineer designing custom UAVs for land surveying. He is a member of the IEEE.