2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)

Local Planning Using Obstacle Estimation and Tangential Escape

Kevin Braathen de Carvalho and Guilherme C. R. de Oliveira and Alexandre S. Brandão

***Abstract*— This work proposes a obstacle estimation algorithm to improve the Tangential Escape Algorithm performance on low cost platforms that have spread sensors. This estimation is done using sensor measurements and linear regression to obtain a first order equation and with its inclination the deviation problem of the Tangential Escape is corrected. Filters are also implemented in order to increase the reliability of the obstacle estimation. Simulations were done to validate the this works proposal using ARIA library integrated with MatLab and MobileSim. Results showed a clear improvement on the path smoothness.**

I. Introduction

Mobile robots is one of the major study fields in robotics and planning is one of its topics which involves autonomous platforms capable of perceiving the environment and taking decisions about their navigation. They should take in consideration the amount of information regarding the capacities of sensing, degrees of freedom of the robots movement as well as the environment around it [1]. To make the robot achieve its goal, several approaches have been proposed, such as global path planning, where the agent knows beforehand information about the environment stored on a map, and then it chooses its path towards its goal. Such approach is classified as deliberative paradigm, and it is discussed in [2]–[9].

This class of algorithms struggle to react to environment changes, and on most real world applications one needs to assume that the agent will deal with a dynamic environment and this makes the reactive navigation. Such behavior is an important part of the robot's autonomy. Many different local planning algorithms have been proposed in the literature such as Potencial Fields Method [10], [11] Virtual Force Field, Vector Field Histogram [12] , Tangential Gap Flow [13], Improved Follow the Gap [14] and Tangential escape [15].

Local planning algorithms excel on what global planning struggles, i.e, they provide efficient tools to real time obstacle avoidance. On the other hand, they sometimes suffer with situations of local minima and consequently they could not find the optimal path towards global navigation. These drawbacks can also be aggravated by sensor's noises, range and area covered at time.

Tangential Escape algorithm is a simple solution to the local planning problem on robotic navigation but it does not work well in platforms with sparse sensors. An algorithm with such low computational cost would greatly benefit low budget platforms that possibly would not be able to afford sensors with high resolution scans. So this work proposes an improvement in the obstacle estimation for usage in the Tangential Escape to suit these platforms. Such improvement might not be useful for high range sensors such as LIDAR or depth sensors.

The manuscript is divided as follows: Section II discusses the Local Planning, explaining how tangential escape works and the modifications are added to it. Section III presents numerical simulations to validate the method as well as discussion about it. Finally, Section IV highlights the main conclusions and some ideas for future works.

Mr. Carvalho thanks CAPES and Department of Informatics, Universidade Federal de Viçosa.

Mr. de Carvalho, M.Sc. Oliveira and Dr. Brandão are with the Núcleo de Especialização em Robótica, Department of Electrical Engineering, Universidade Federal de Viçosa, Viçosa - MG, Brazil. E-mail: kevinbdc@gmail.com, guilherme@ufv.br, alexandre.brandao@ufv.br

II. Local Planning

The proposed algorithm performs obstacle estimation in order to aid the tangential escape in the definition of the closest robot-obstacle distance as well as its orientation from the robot's reference frame. This section details how the original algorithm works and how its behavior can be improved when a set of spread sensors is used.

A. Tangential Escape

Tangential Escape presents a low computational cost due to its mathematical simplicity, which is highly appreciated in low cost platforms. Despite having solid characteristics, it may struggle on robots with highly spread sensors, because the angle of the closest reading could not represent the actual closest distance to the obstacle. Thus to overcome this problem, the present paper proposes a way to improve the obstacle estimation, as shown hereinafter.

The strategy consists in a simple math operation that takes in account the closest sensor measurement to the obstacle and the angle of its reading, whose current reference is the robot's axis. In others words, the method is based on modifying the robot's current goal by creating a new virtual one, which is tangent to the obstacle, as one can see in Figure 1.

The angle γ defines how the real goal should rotate to be tangent (or parallel) to the obstacle. Notice in Figure 1 that the virtual target is a viable way of overcoming the obstacle while minimizing the risk of collision or even avoiding getting stuck in it.

The rotation angle γ is given by

$$\gamma = \begin{cases} -\frac{\pi}{2} + \beta - \alpha, & \text{if } \beta \geq 0 \\ +\frac{\pi}{2} + \beta - \alpha, & \text{otherwise,} \end{cases} \tag{1}$$

978-1-5386-7761-2/18/$31.00 ©2018 IEEE
DOI 10.1109/LARS/SBR/WRE.2018.00062

where β is the angle of the closest robot-obstacle sensor measurement (d_{min}), and α is the angle between the real goal and the robot heading.

It is worthy mentioning that Tangential Escape strategy is active whenever the closest robot-obstacle sensor measurement d_{min} is lower than a present distance d_{obs}, which is the limit of the safety zone. Thus such zone is an area without collision risk, whenever the robot navigates outside it.

Analyzing Equation 1, one can see that a wrong β would lead to a virtual goal that won't be tangent to the obstacle, making the path lead the robot further away or, even worse, closer to the obstacle. This happens on most cases using a platform with sparse sensors, since it has fixed angles on its readings, so if the closest distance is between two of its sensors the provided β will certainly be wrong as its shown on Figure 2.

Algorithm 1 describes the Tangential Escape strategy. First, the virtual goal is defined as the current goal. If the minimum distance (d_{min}) is smaller than the observed one (d_{obs}), the angles β, γ and α are calculated. After that γ is computed according to the obstacle position relative to the robot. If β is negative, the obstacle is on the right-side of the robot. On the other hand, if β is positive, the obstacle is on the left-side of the robot. In the sequence, the distance between the current robot's position and the real goal is used to determine the virtual goal. In contrast, if d_{min} is bigger than d_{obs}, γ's value is smoothly reduced using the forget factor. Finally, the virtual goal's coordinates, rotated or not, return as the output.

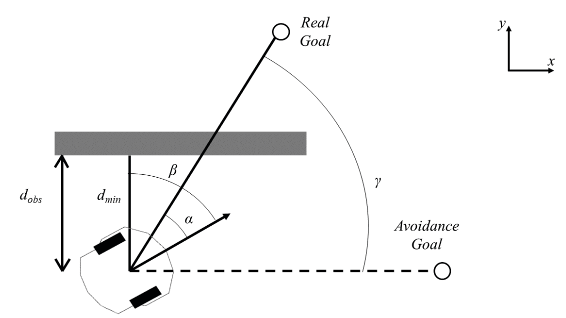


Fig. 1. Tangential Escape.

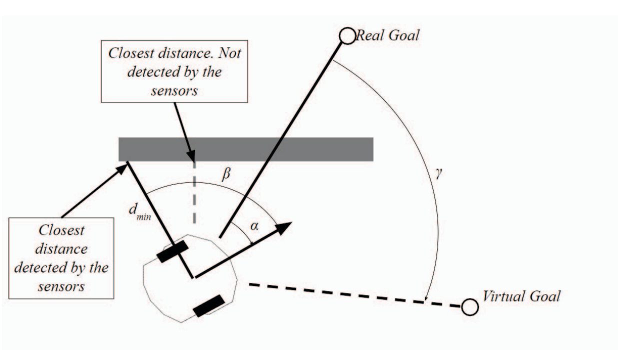


Fig. 2. Tangential Escape with wrong β.

Algorithm 1 Tangential Escape Algorithm.

Require: Input: fe, d_{min}, d_{obd}
Ensure: Output: Virtual Target Position X_v

$X_v \leftarrow X_d$
if $d_{min} < d_{obs}$ **then**
  $\beta \leftarrow ClosestSensorMeasurementAngle$
  $\theta \leftarrow \arctan(y_g/x_g)$
  $\alpha \leftarrow \theta - X_{yawl}$
  if $\beta < 0$ **then**
    $\gamma \leftarrow -\pi/2 - \alpha + \beta$
  else
    $\gamma \leftarrow +\pi/2 - \alpha + \beta$
  end if
  $d \leftarrow norm(X_v - X)$
  $X_v \leftarrow X + \tanh(d) \begin{bmatrix}\cos(\theta-\gamma) & \sin(\theta-\gamma)\end{bmatrix}^T$
else
  $\gamma \leftarrow \gamma(1 - fe)$
end if
return X_v, γ

It is important to highlight two points in the algorithm. First, hyperbolic tangent function saturates the virtual goal, which makes the robot navigate slower because the new destination is close to it. Second, the forget factor f_e guarantees a smooth transition from deviating to regular navigation.

B. Obstacle Estimation

In order to use Tangential Escape as better as possible, the robot must have a good sensor scan, so it can always provide the actual β. Pioneer 3-DX is the robot used in the numerical simulation. It has 8 sonar sensors spread around it on fixed angles [16]. Thus it is possible that β could not represent the angle of the closest distance from the robot to the obstacle, as illustrated in Figure 2, in other words, the virtual goal is not correctly determined. So, this work proposes a way of getting β using virtual lines that partially represents the obstacle.

To accomplish it, the first step is the conversion of the sensor measurements to Cartesian coordinates in the robot frame, and then in the world fixed frame. After that, for each three consecutive sonar measurements, a linear regression is done and a first order equation is defined, which is now labeled as *virtual obstacle*. Finally, the orthogonal line between the robot and the virtual obstacle is defined, a estimated β is computed, and then it is applied to the Tangential Escape strategy.

Figure 3 illustrates the sonar measurements of the Pioneer 3-DX robot, in light-gray color. In our case, only the eight frontal sensors are available, located at -90°, -50°, -30°, -10°, 10°, 30°, 50° and 90°. Notice they are spread and thus they could not observe an obstacle (mainly L- or V-shape ones).

For each set of three measurement one can estimate a first-order equation. For instance, one line can be estimated

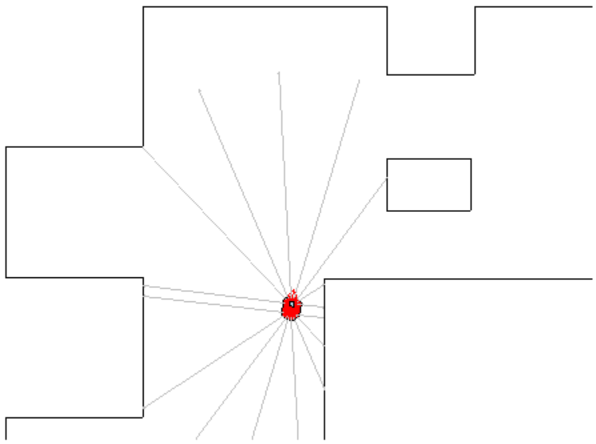


Fig. 3. Environment with obstacles and walls.

using sensors located at -90°, -50° and -30°, and another one, using -50°, -30° and -10°, and so on. However some line could not represent an obstacle of interest, so they must be discarded, as shown in the sequence.

C. Obstacle Filtering

Once knowing the sensor reading are corrupted by noises, the obstacle detection is compromised, resulting in poor estimation of lines that represent an obstacle. To assure a good enough approximation of the first-order equation, its coefficient of determination R^2 is evaluated according a predetermined threshold, as described below.

Whenever a sensor returns its maximum range, one can infer there is no obstacle in that direction. Thus such measurement is ignored during the line estimation. In other words, if the measurement is greater than 3 meters, such information is discarded for the *virtual obstacle* estimation.

Another way to discarded line estimation is analyzing θ angle and its expected value. From Figure 4, S_1 and S_2 are two consecutive measurements, and ϕ is the angle between them, so one gets

$$\frac{S_1}{\cos(\theta+\phi)} = \frac{S_2}{\cos\theta} \tag{2}$$

from sine law. Assuming $\theta = 15°$, all S_2 measured distance given by the sonar is discarded from the obstacle estimation. In other words, if S_2 is greater than the expected value, we assume it is an erroneous sonar reading.

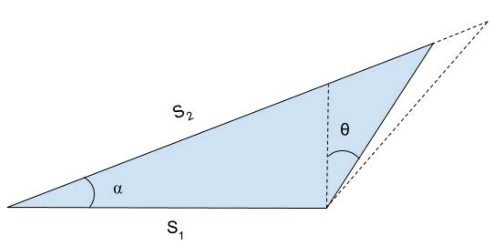


Fig. 4. Obstacle filtering.

III. Simulation Results and Discussion

In order to compare the navigation performance from the regular Tangential Escape and the improvements on β estimation, numerical simulations are run in *MobileSim* and *Matlab*, using ARIA Library compiled in MEX to link them.

To validate the proposal three scenarios are used. All of them has corner obstacles, which can be associated with long and short walls, large and narrow corridors, small and big isolated obstacles, in a real situation. The representation of these scenarios are shown in Figures 5(a), 5(c) and 5(e).

For all illustrated environment, the robot's task is to achieve a desired goal and then return to the starting point, avoiding obstacles. It is important to highlight the robot does not have any prior information about the scenarios and it does not store data about the obstacle during its displacement, therefore it navigates in a purely reactive way. Table I presents the constants adopted for all numerical simulations.

TABLE I

The adopted constants for numerical simulations.

Constant	Value
R^2	0.8
f_e	0.9
d_obs	1m

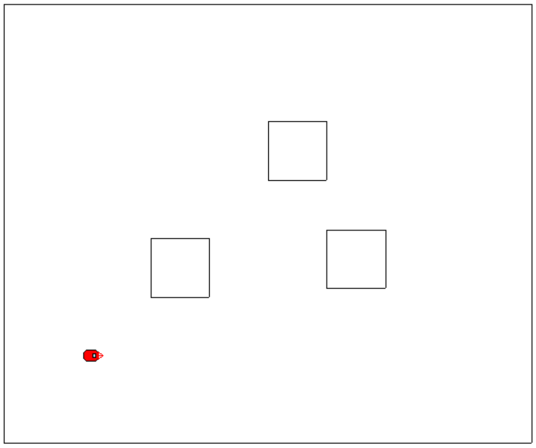
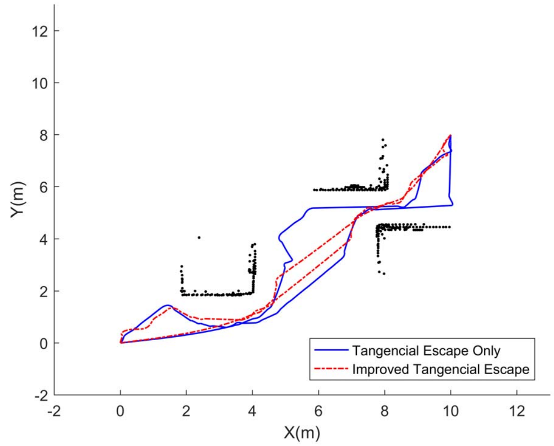
The robot's traveled path are shown in Figures 5(b), 5(d) and 5(f), where the blue solid line represents the standard Tangential Escape strategy and the red dotted line represents the Tangential Escape with obstacle estimation algorithm. The black dot points represent the obstacles as they are "seen" by the robot from its sonar measurements.

In the first simulation, the robot should avoid few small obstacles. When the pure Tangential Escape algorithm is used, the robot deviates from one obstacle to another, navigation very close to them, as one can verify in Figure 5(b). In contrast, when the obstacle is estimated, the robot performs a smooth displacement, saving time and energy, besides traveling a safer path.

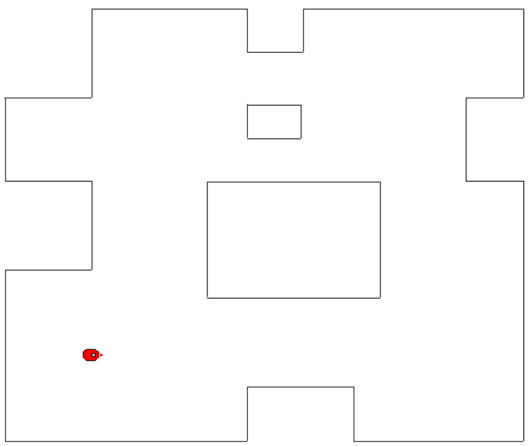
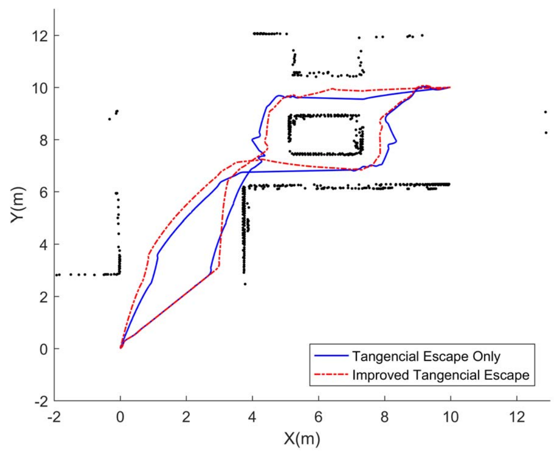
Notice that both strategies have heading oscillations when the robot faces the first obstacle. However, two distinct behaviors can be observed after overcoming it, the obstacle estimation provides a smoother and an improved path. The robot navigation is available here.

In the second simulation it is important to highlight when the robot deviates from the middle obstacle (the island one). In such a case, when the robot faces such obstacle and its edges, depending of the closet distance and the relative β value, the avoidance maneuver may be to the right- or left-side. Moreover, after starting deviating, this obstacle becomes similar to a narrow corridor and the tangential escape switches among the closest distances, which can be on robot's right- or left-side. The robot navigation in the second scenario is available here.

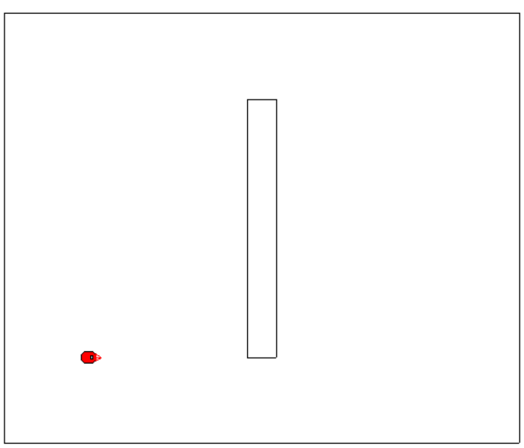
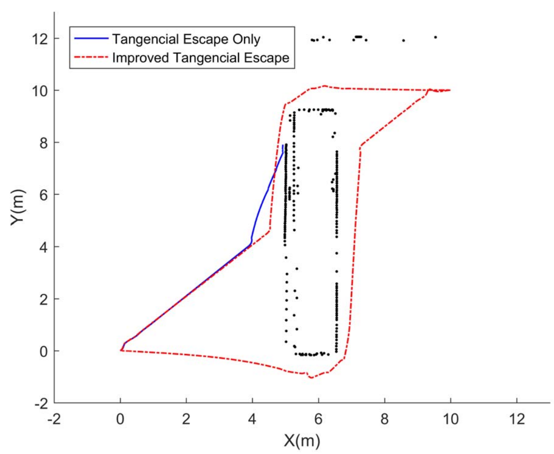
Notice a smooth path is taken by the obstacles estimation aiding the Tangential Escapes. In summary, the slight inclination towards the target is severely reduced and the

(a) (b)

(c) (d)

(e) (f)

Fig. 5. Simulation scenarios.

robot's oscillations when it is faced an obstacle corner is practically removed. It is worthy pointing out R^2 analyzes how the strategy can work in real situation, once higher noises may be presented in the distance measurements, and it can compromise the line estimation.

The third simulation shows one of the biggest struggles that a platform with spread sensors has using Tangential Escape strategy. In this kind of obstacles, β feeds the algorithm is a wrong way. The rotation of the virtual goal is not accurate and the robot slowly drifts toward the target. However the bigger the obstacle the higher the collision risk.

In this simulation when the pure Tangential Escape is used, the robot collides on the way from the start point to the target. In contrast, using the obstacle estimation, the Tangential Escape is able to overcome the obstacle and to accomplish the mission safely. The robot navigation in the third scenario is available here.

The simulation time considering the tangential escape with and without obstacle estimation is shown in Table II. Notice the time spent to accomplish the tasks is smaller in both case in which the robot accomplishes the mission. Therefore, the obstacle estimation aids the Tangential Escape strategy, and consequently it helps to robot to fulfill its mission.

TABLE II

THE SIMULATION TIME CONSIDERING THE TANGENTIAL ESCAPE WITH AND WITHOUT OBSTACLE ESTIMATION.

Map	Pure Algorithm	Aided Algorithm
1	175s	130s
2	187s	168s

It is important to say the robot is not able to accomplish the mission in the third scenario. Sensor noises and lack of measurements are some reasons for the robot collision with long walls (or corridors). In these environments, the measurements are not reliable once the sonar signal could not be correctly reflected. In turn, even for reliable measurements, the robot can collide due to the low number of sonars available, since β can only assume eight different values. Such limitation does not happen in our proposal, because $\beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ can assume any value in this interval.

IV. Concluding Remarks

This paper proposes an obstacle estimation algorithm to aid the Tangential Escape strategy to better suit its use in low-cost platforms with limited and sparse distance sensors. Numerical simulations validate the proposal. In three different scenarios, the robot must perform position tasks, going to a desired target and returning to the starting point, avoiding obstacles. The results illustrate that estimating obstacles helps the robot run a smooth path and spend less time to accomplish the task.

References

[1] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[2] J. L. Jones, A. M. Flynn, and B. A. Seiger, *Mobile robots: inspiration to implementation*. AK peters Wellesley MA, 1999, vol. 2.

[3] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[4] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[5] ——, "Sampling-based optimal motion planning for non-holonomic dynamical systems," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5041–5047.

[6] M. Brand, M. Masuda, N. Wehner, and X.-H. Yu, "Ant colony optimization algorithm for robot path planning," in *Computer Design and Applications (ICCDA), 2010 International Conference on*, vol. 3. IEEE, 2010, pp. V3–436.

[7] P. Das, H. S. Behera, and B. K. Panigrahi, "A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning," *Swarm and Evolutionary Computation*, vol. 28, pp. 14–28, 2016.

[8] L. Juntao, D. Tingting, L. Yuanyuan, and H. Yan, "Study on robot path collision avoidance planning based on the improved ant colony algorithm," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016 8th International Conference on*, vol. 2. IEEE, 2016, pp. 540–544.

[9] S. Alnasser and H. Bennaceur, "An efficient genetic algorithm for the global robot path planning problem," in *Digital Information and Communication Technology and its Applications (DICTAP), 2016 sixth international conference on*. IEEE, 2016, pp. 97–102.

[10] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings of the 1991 IEEE international Conference on Robotics and Automation*. IEEE, 1991, pp. 1398–1404.

[11] M. C. Lee and M. G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," in *Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, vol. 2. IEEE, 2003, pp. 735–740.

[12] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE transactions on robotics and automation*, vol. 7, no. 3, pp. 278–288, 1991.

[13] M. Demir and V. Sezer, "Improved follow the gap method for obstacle avoidance," in *Advanced Intelligent Mechatronics (AIM), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1435–1440.

[14] M. Mujahed, D. Fischer, and B. Mertsching, "Tangential gap flow (tgf) navigation: A new reactive obstacle avoidance approach for highly cluttered environments," *Robotics and Autonomous Systems*, vol. 84, pp. 15–30, 2016.

[15] A. S. Brandão, M. Sarcinelli-Filho, and R. Carelli, "An analytical approach to avoid obstacles in mobile robot navigation," *International Journal of Advanced Robotic Systems*, vol. 10, no. 6, p. 278, 2013.

[16] "Pioneer 3dx datasheet," http://www.mobilerobots.com/Libraries/Downloads/Pioneer3DX-P3DX-RevA.sflb.ashx, 20017, [Online; accessed 19-July-2017].