# Holonomic planar motion from non-holonomic driving mechanisms: The Front-Point Method

Selim Temizer and Leslie Pack Kaelbling

Massachusetts Institute of Technology
Artificial Intelligence Laboratory
Cambridge, Massachusetts, USA

## ABSTRACT

There are many methods that address navigation and path planning for mobile robots with non-holonomic motion constraints using clever techniques and exploiting application-specific data, but it is always better not to have any such constraints at all. In this document we re-examine the capabilities of some popular driving mechanisms from a different perspective and describe a method to obtain holonomic motion using those mechanisms. The main idea is to not concentrate on the center of the driving mechanism (which is the usual choice) as the reference point for our calculations, but to select another point whose motion in the x-y plane is not constrained in any direction, and which is also a logical and useful substitute for the center. In addition to the derivation of the forward and inverse kinematics equations for the new reference point, we also explain how to further simplify the design of a controller which uses the described method to compute motion commands for the robot. In order to illustrate the ideas, we present graphs that were plotted using the actual parameter values for a synchronous-drive research robot.

**Keywords:** Mobile robots, holonomic motion, non-holonomic motion constraints, local navigation

## 1. INTRODUCTION

The mechanics of most real robots and vehicles do not let them move freely in any direction instantaneously. For example, due to the limitation of the steering mechanism of the cars (ackerman steering), immediate sideways motion is impossible, and drivers have to go through complex maneuvers for parking. These kind of constraints on motion are called *non-holonomic motion constraints*. Latombe[1] gives a formal definition and detailed analysis of non-holonomic constraints, and more information about steering mechanisms, non-holonomic constraints and mobile robot hardware can be found in the book written by Dudek and Jenkin.[2]

In this document, we will sometimes refer to our research robot Erik—a Real World Interface (RWI) Model B21r indoor robot—and illustrate the ideas with graphs that are plotted using Erik's parameter values. Erik is a cylindrical robot. He has a synchronous-drive mechanism which also imposes non-holonomic constraints on his motion.

On the other hand, some of the powerful navigation methods like the potential field method[3] are best suited for holonomic bodies, because they require non-constrained motion in 2D coordinates. There are two alternatives that can be followed: We can either design our navigation algorithms with the mechanic limitations of the robots in mind, or we can try to find some other techniques for robots to overcome those motion constraints.

Some examples of the first approach are as follows: Polynomial-time algorithms for producing paths with lengths that are either optimal or within an additive constant of optimal are described by Agarwal et al..[4] Methods using harmonic functions and a resistive 3D grid analogy are offered by Connolly and Grupen.[5] Piecewise linear paths are used by Arkin[6] for the actual robot experiments of some navigation techniques, and Simmons[7] formulate the problem as one of constrained optimization in velocity space, where the velocity space is a direct representation of all of the motion capabilities of the robot. The same velocity space approach is also used

Selim Temizer: E-mail: temizer@ai.mit.edu, Phone: +1 617-253-1611.
Leslie Pack Kaelbling: E-mail: lpk@ai.mit.edu, Phone: +1 617-258-9695, Fax: +1 617-253-7781.

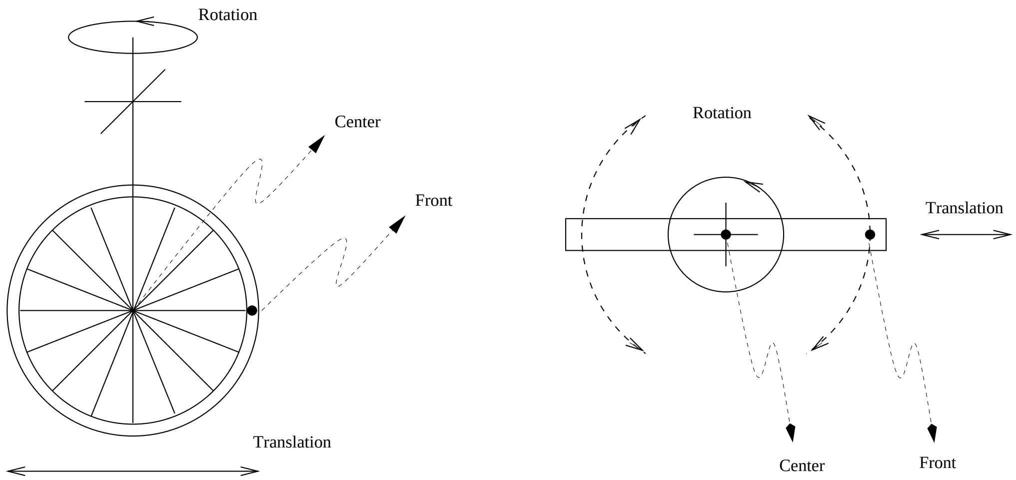



**Figure 1**: Side view (left) and top view (right) of a bicycle wheel.

by Fox et al.[8] for reactive collision avoidance for mobile robots equipped with synchronous-drive mechanisms. Solutions that follow the second approach usually first create free paths ignoring the non-holonomic constraints, and then keep the robot aligned to the path by frequent reorientations. More examples and state-of-the-art techniques in non-holonomic path planning are compiled by Gupta and Popil.[9]

The method that we describe in this document offers a solution to the navigation problem by following the second approach. If we change our reference point for the path-following task, it becomes possible to use some popular driving mechanisms to generate non-constrained motion for the new reference point, which implies that the new reference point can follow any given path without the need for reorientations to stay on the path, or without modifying the path itself.

In the following section, we will talk about the motivation behind, and the basic idea of the front-point method. In Sect. 3, we will discuss how to choose the new reference point and the advantages and disadvantages of changing the reference point. Derivation of the forward and inverse kinematics equations for the new reference point will be presented in Sect. 4. We will also show how to compute the maximum attainable speed in Sect. 5. We want to note that throughout this document, we will describe the front-point method and demonstrate its application from the perspective of synchronous-drive mechanisms. However, the basic idea behind the method is very simple and can also be applied to differential-drive mechanisms and possibly to some other drive mechanisms. We will also cover the application of the method to differential-drive mechanisms in Sect. 6. Section 7 will conclude the document by highlighting some important points.

## 2. MOTIVATION AND BASIC IDEA

One of the driving mechanisms used in today's research robots, the synchronous-drive mechanism (also known as synchro-drive mechanism), can be thought of as being equivalent to a bicycle wheel when the basic functionality is considered. Both systems are capable of moving forward, moving backward and rotating around the central axis. Both systems also have a heading direction, and therefore a notion of front and rear. Figure 1 shows how a bicycle wheel moves (think of the front wheel).

The motion of the center of the wheel is constrained in the sense that it cannot move instantaneously in all possible directions. For example, the wheel cannot move sideways, hence the center cannot move sideways directly. The center is a typical example of a non-holonomic body, and in order to move it sideways, some non-simple maneuvers of the wheel are needed.

As mentioned before, a synchro-drive robot, such as Erik, also moves in a similar fashion to a bicycle wheel. It can only translate forward, or backward, or it can rotate around its central axis. The center of the robot is

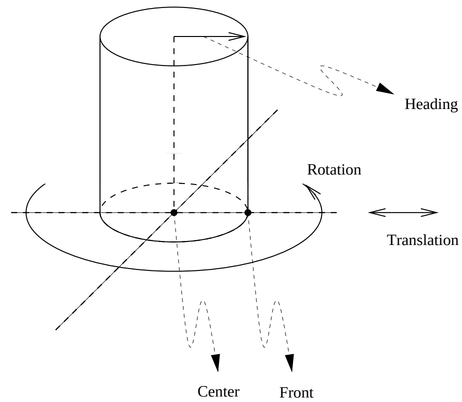



**Figure 2**: Cylindrical synchronous-drive robot.

again a non-holonomic body in the x-y plane. Figure 2 shows the available instantaneous motions for Erik's center.

Given a starting position in the x-y plane and an initial heading for Erik, let's say that his task is to reach a target position and orientation. The non-holonomic motion constraint for a synchro-drive robot is actually not a very strict constraint, and as an example, we can first rotate Erik until he faces the target, and then move him forward until his center is located directly above the target position. Lastly, we stop him and rotate him again to attain the desired target orientation. In this simple and trivial solution, the robot also moves in a straight line (traverses the shortest path) towards the target, which is usually the desired way of navigation.

But sometimes we would like to have our robot move in any arbitrary direction freely and instantaneously. For example, if we want to use some navigation methods based on potential field models for obstacles and targets, we provide the coordinates of our robot as the input, and the method returns us a certain direction to move in. Depending on the orientation of our synchro-drive robot, most of the time we will not be able to move in the desired direction immediately. We should first rotate the robot for the correct heading before the translation in the desired direction can begin. Aligning the heading in each iteration of a navigation method's controller loop then becomes a tedious task and results in a non-smooth motion (it also increases the amount of drift in the robot's odometry system if the robot has one on-board).

Having a non-holonomic body also leads to complex controller designs. The robot may need to do some intensive planning to achieve even some fairly simple tasks since it cannot move freely. On the other hand, the controller design for a holonomic body is very simple and easy, because immediate motion in all directions is possible, and there is no need to do planning for avoiding obstacles that may end up being on the path of the robot just because the non-holonomic robot cannot move in a direct, obvious and collision-free path.

The front-point method is a re-examination of the capabilities of some popular driving mechanisms from a different perspective. It is a technique that converts the non-holonomic robot into a holonomic body. The main idea is to not concentrate on the center of the robot as the reference point for navigational calculations, but to select another point, whose motion in the x-y plane is not constrained in any way, and which is also a logical and useful substitute for the center. Using the center of the robot as the reference has many beneficial properties as in the case of configuration-space planners[10] and many other planning algorithms. But if the cost of changing the reference system is less than the cost of dealing with non-holonomic constraints, then the method described here may be of substantial use for the application at hand.

The front-point method can easily be implemented in a very tiny code and applied on synchronous-drive mechanisms (and some other mechanisms as well). Also the execution of this small piece of code does not involve extensive mathematical computations. Therefore, having this method as a subprogram inside the controller for a synchro-drive robot may help the robot achieve some otherwise very difficult maneuvers easily and efficiently (both in terms of time and energy, since the new reference point can reach the target location for the robot by traversing the shortest distance).

## 3. SELECTION OF NEW REFERENCE POINT, ADVANTAGES AND DISADVANTAGES

As we will see the mathematical details in the following sections, it turns out that any point that does not lie on the main axis of the driving mechanism qualifies to be selected, since the non-holonomic motion constraints only apply to the points on the main axis. We can pick any qualifying point and be sure that it can move in any direction instantaneously. But a logical and a useful selection is the front point, i.e., the front of a cylindrical robot (or the front of the bicycle wheel since they are equivalent motion systems), because it is one of the distinguished points of the robot as it determines the heading, and also we usually refer to the robot (and also to us - humans) "reaching the target" when the front point comes in contact with, or comes very close to the target. (As opposed to the hypothetical point targets specified by pairs of Euclidian coordinates on the x-y plane, the robot can never place its center above physical target locations - since this implies a crash into the target.)

The main advantage of converting a non-holonomic body to a holonomic body is that the controller design for a holonomic body is very simple, and any high-level or low-level optimal navigation-planning algorithm can easily be used without any modifications.

One disadvantage of selecting the front point (or any qualifying point) as our new reference is that in order to move the front point in certain directions with a constant speed, it may be necessary to apply very high rotation and translation speeds depending on the desired motion speed of the front point. This means that the center may need to accelerate very quickly if high speeds are desired for the front point motion. Another disadvantage is the following: The maximum speed that the front point can attain at any time depends on the robot's heading, which makes the controller programming non-uniform. But we can still program our controllers without dealing with the fact that the maximum speed is not constant. We will discuss how to compute a maximum speed value that is guaranteed to be available independent of the robot's heading.

Actually, we can have this conversion method as a subprogram for our robot, and use it whenever it is appropriate for us and makes things easier. For all other situations, we can continue to live with our robot's inherent non-holonomic constraints, and use the algorithms we used to run before. So, there are actually no disadvantages to having this method at hand as an additional navigation tool.

## 4. KINEMATICS EQUATIONS FOR THE NEW REFERENCE POINT

Even though we mentioned that the front point is a good selection as the new reference point, in this section, we will derive our equations more generally for an arbitrary reference point.

For a synchronous-drive robot, the control commands are the translation speed $\vec{V}$ (magnitude $V$), and the angular rotation speed $\vec{\omega}$ (magnitude $\omega$, counter-clockwise rotation is positive). The other terms that we will use are explained in Table 1 and illustrated in Figure 3. For the robot heading $\theta$, positive x direction is 0 radians, and counter-clockwise rotation is positive.

Given the translation speed $V$ (in the direction of robot's heading), the rotation speed $\omega$ (in counter-clockwise direction), the heading of the robot $\theta$, and the polar coordinates $r$ and $\beta$ of the reference point with respect to the center, the reference point will have a speed of $\vec{V_f} = \langle \dot{x_f}, \dot{y_f} \rangle$ (where $\dot{x}$ represents $\frac{dx}{dt}$). The forward kinematics equations are:

$$\dot{x_f} = V\cos\theta - (r\omega)\sin(\theta+\beta),$$

$$\dot{y_f} = V\sin\theta + (r\omega)\cos(\theta+\beta),$$

$$\dot{\theta} = \omega.$$

In the above equations, $(r\omega)$ is the instantaneous speed (due to rotation) at the reference point in the direction that is perpendicular to the line connecting the center and the reference point. Written in matrix notation, the reference point forward kinematics equation is:

$$\begin{bmatrix} \dot{x_f} \\ \dot{y_f} \end{bmatrix} = \begin{bmatrix} \cos\theta & -r\sin(\theta+\beta) \\ \sin\theta & r\cos(\theta+\beta) \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix} = A \begin{bmatrix} V \\ \omega \end{bmatrix}.$$

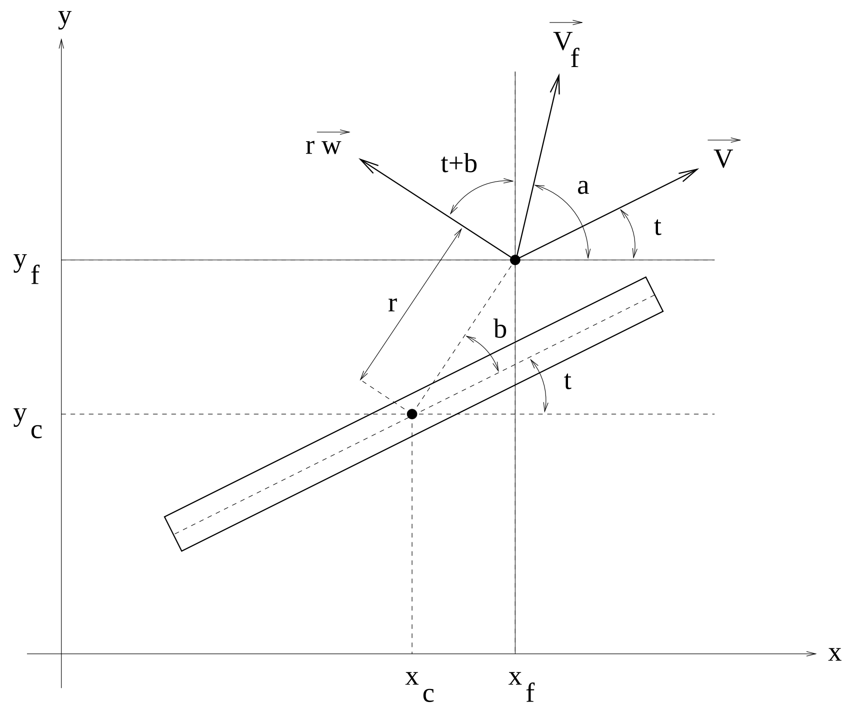



**Figure 3.** Illustration of the front-point idea for synchronous-drive mechanism using the bicycle wheel example. In the figure, the variables $\alpha$, $\beta$ and $\theta$ are represented by the letters a, b and t respectively.

**Table 1**: Explanation of the terms for synchronous-drive mechanism.

| **Explanation** | **Term** | **Value for Erik** |
|---|---|---|
| Robot heading angle | $\theta$ | N/A |
| Direction of motion of reference point | $\alpha$ | N/A |
| Polar coordinates of reference point | $\beta$ and $r$ (with respect to the center) | N/A |
| Robot radius | $r$ (if reference point = front point) | 0.267 meters |
| Minimum translation speed | $V_{min}$ | -0.9 meters/sec |
| Maximum translation speed | $V_{max}$ | 0.9 meters/sec |
| Minimum rotation speed | $\omega_{min}$ | -2.915 radians/sec |
| Maximum rotation speed | $\omega_{max}$ | 2.915 radians/sec |

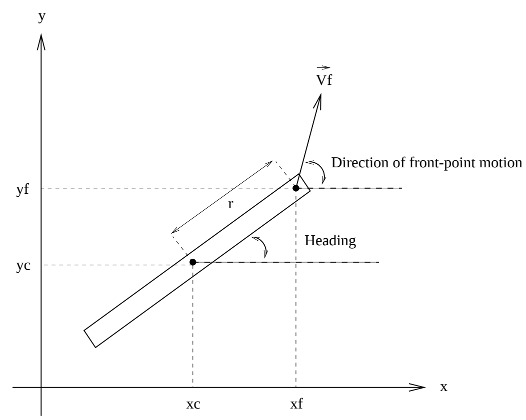



**Figure 4**: Illustration of the problem with the bicycle wheel example.

The determinant of the matrix $A$ is $(r\cos\beta)$. Note that the determinant is zero only if $r = 0$ or $\cos\beta = 0$. So, any point on the main axis (the line that is perpendicular to the bicycle wheel and intersecting it at the center) does not qualify to be selected as our new reference, because points on the main axis are non-holonomic. To find the inverse kinematics equations for the reference point, we can just invert matrix $A$ and obtain the result directly:

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = A^{-1} \begin{bmatrix} \dot{x_f} \\ \dot{y_f} \end{bmatrix} = \begin{bmatrix} \cos(\theta+\beta)/\cos\beta & \sin(\theta+\beta)/\cos\beta \\ -\sin\theta/r\cos\beta & \cos\theta/r\cos\beta \end{bmatrix} \begin{bmatrix} \dot{x_f} \\ \dot{y_f} \end{bmatrix}. \quad (1)$$

Therefore, given a desired velocity $\vec{V_f} = \langle \dot{x_f}, \dot{y_f} \rangle$ for the reference point, in order to find the required translation and rotation commands ($V$ and $\omega$) we can use Equation (1) which only depends on the heading angle $\theta$ and the polar coordinates $r$ and $\beta$ of the reference point.

## 5. COMPUTATION OF THE MAXIMUM SPEED ATTAINABLE BY THE FRONT POINT

In this section, in order to concentrate on the main idea, we will simply choose the reference point to be the front point (i.e., $\beta = 0$, $r$ = radius of robot) and address the following problem: Given the heading, radius and the speed limits of the robot, move the front point in a specific direction which is $\alpha$ degrees from the positive x axis, with maximum speed. We should find the correct values for $V$ and $\omega$ to achieve the desired task. The situation is illustrated in Fig. 4 with the bicycle wheel example.

This is a simple maximization problem which can be presented as:

$$\text{maximize} \left[ \sqrt{\dot{x_f}^2 + \dot{y_f}^2} \right]$$

with respect to $V$ and $\omega$, and subject to the constraints:

$$\dot{x_f} = V\cos\theta - \omega(r\sin\theta),$$

$$\dot{y_f} = V\sin\theta + \omega(r\cos\theta),$$

$$\dot{y_f}/\dot{x_f} = \tan\alpha,$$

$$\omega_{min} \le \omega \le \omega_{max},$$

$$V_{min} \le V \le V_{max}.$$

In order to visualize the solution more easily, we can also present this problem as a simple linear combination problem: We have two vectors $\vec{V}$ and $r\vec{\omega}$ ($V_{min} \le V \le V_{max}$, $\omega_{min} \le \omega \le \omega_{max}$, and r is a constant scalar), and our aim is to choose the appropriate magnitudes for these two vectors in the allowed ranges such that their sum $\vec{V_f} = \vec{V} + r\vec{\omega}$ will have a specified orientation and the largest possible magnitude.

The solution to this problem is very simple and the tip of the $\vec{V_f}$ always lies on the edges of a rectangle containing all possible allowed linear combinations of the given vectors as can be seen in the polar graph in Fig. 5 which was plotted for Erik using his actual parameter values.

As we mentioned before, the maximum attainable speed is not constant, and depends on the values of $\alpha$ and $\theta$. This may seem to be a problem, because while writing actual controller programs, instead of trying to find the maximum speed for each case we encounter, we may wish to have a maximum speed value that is guaranteed to be available for any combination of $\alpha$ and $\theta$. This value is easy to obtain, and is the minimum of all the maximum attainable speeds for all possible headings and directions. Therefore, we can compute this value once for our robot, and then use it for all cases, with the guarantee that the robot will never try to exceed its speed limits. This value, which we can use as a constant maximum guaranteed speed in all of our controller designs, is the following:

$$V_{constant} = \min\left( |V_{max}|, |V_{min}|, |r\omega_{max}|, |r\omega_{min}| \right).$$

As an example, the two graphs in Fig. 5 show both the maximum attainable speeds, and the constant maximum speed that is always available for Erik for any $\alpha$ direction with a fixed $\theta = 0$. It should be noted that there is no loss of generality by fixing $\theta$ in the given example, because an increase in $\theta$ simply results in a phase shift of the graph values, and the phase shift amount is exactly equal to $\theta$. To see this clearly, refer to the graphs in Fig. 6 which are computed for $\theta = 30$ degrees with all other parameters being the same.

Therefore, we can easily use the pre-computed constant maximum speed value in designing the controllers. One disadvantage of using a constant value is of course wasting the extra performance available for most of the situations, but this is one of the many trade-offs between ease of design and using extra performance in many design problems.

## 6. APPLICATION TO DIFFERENTIAL DRIVE MECHANISMS

Differential drive is a very simple drive mechanism consisting of two wheels mounted on a common axis. The wheels are controlled by separate motors. Most differential drive mechanisms also use castor wheels for stability. A differential drive robot controls its pose by providing independent velocity control to the left and right wheels. For a differential-drive robot, the control commands are the left and right wheel angular velocities $\vec{\omega}_l$ and $\vec{\omega}_r$ (with magnitudes $\omega_l$ and $\omega_r$, respectively).

As it was the case for the synchronous-drive mechanism, if we try to solve the inverse kinematics equations for the center of the drive system, which is the center of the axle, we will again face non-holonomic motion constraints. For example, immediate sideways motion is not possible for the center of this mechanism either. In this section, we will explain how to apply the front-point method to differential-drive mechanisms, by deriving the forward and inverse kinematics equations for an arbitrary reference point. The explanation of the terms we will use are given in Table 2 and an illustration is given in Figure 7.

Applying an angular velocity to the right wheel ($\vec{w_r}$) produces an instantaneous velocity $\vec{V_r}$ at the reference point with direction perpendicular to the line connecting the reference point to the center of the left wheel, and with magnitude $V_r = \frac{d_1}{2l}(rw_r) = \left(\frac{d_1 r}{2l}\right) w_r$. Similarly, $\vec{w_l}$ will produce $\vec{V_l}$ with magnitude $V_l = \frac{d_2}{2l}(rw_l) = \left(\frac{d_2 r}{2l}\right) w_l$. The fixed value $r$ is the radius of the wheels and $2l$ is the length of the axle. Note that $d_1 = 2l \sin\beta / \sin(\alpha+\beta)$ and $d_2 = 2l \sin\alpha / \sin(\alpha+\beta)$. When we apply the angular velocities $\vec{w_r}$ and $\vec{w_l}$, the reference point will start moving with a linear velocity $\vec{V_f} = \langle \dot{x_f}, \dot{y_f} \rangle$.

$$\dot{x_f} = V_r \cos(\theta+\alpha) + V_l \cos(\theta-\beta) = \omega_r \left[ r d_1 \cos(\theta+\alpha)/2l \right] + \omega_l \left[ r d_2 \cos(\theta-\beta)/2l \right],$$
$$\dot{y_f} = V_r \sin(\theta+\alpha) + V_l \sin(\theta-\beta) = \omega_r \left[ r d_1 \sin(\theta+\alpha)/2l \right] + \omega_l \left[ r d_2 \sin(\theta-\beta)/2l \right].$$

In matrix notation, the forward kinematics equation for the reference point is:

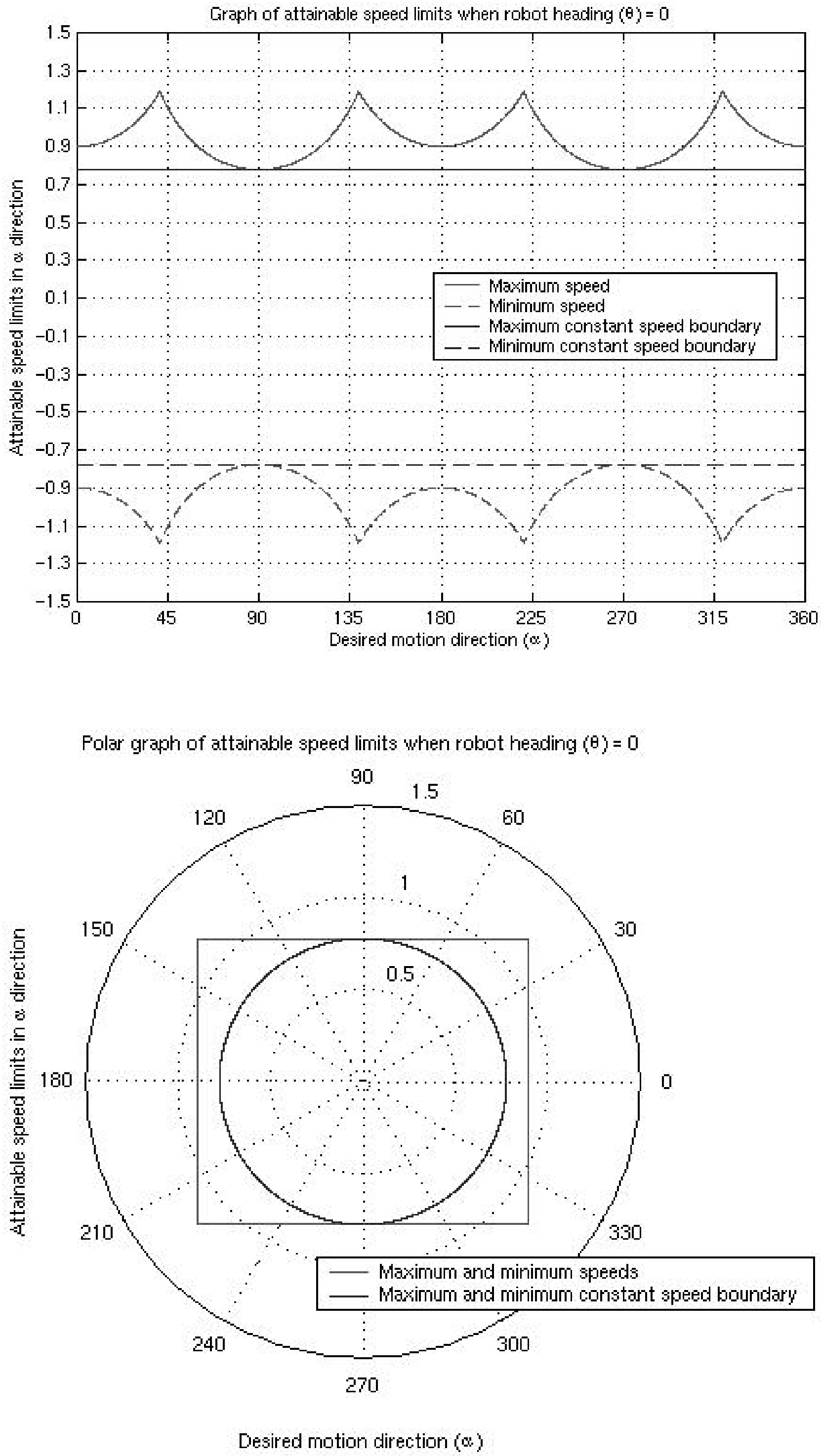
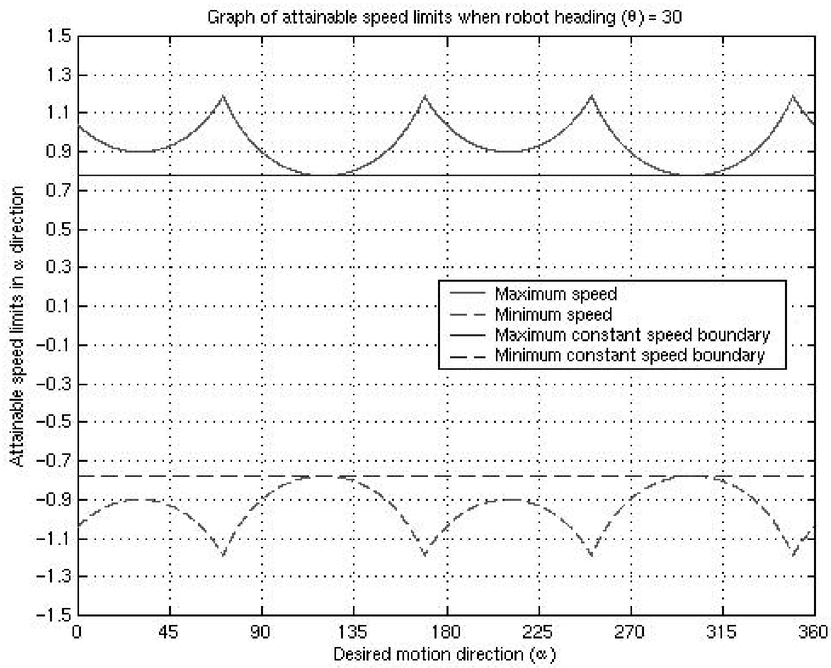



**Figure 5.** In the graph at the top, the straight lines show the limits of the constant speed, and the curves show the actual speed limits. In the polar graph at the bottom, the available constant speed is bounded by the circle. The actual limits lie on the edges of the rectangle.



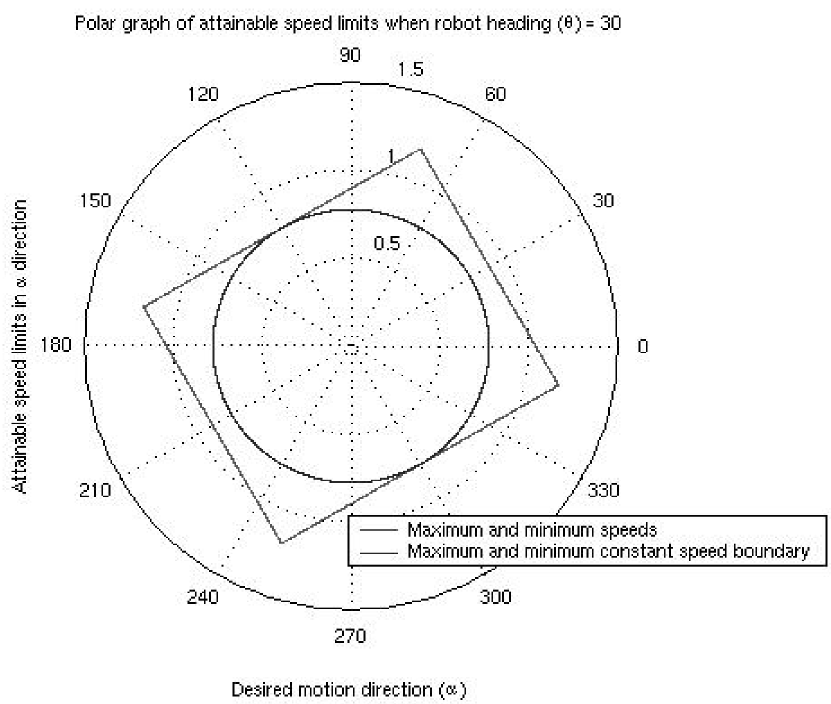




**Figure 6**: Limits of attainable speeds with a 30° phase shift.

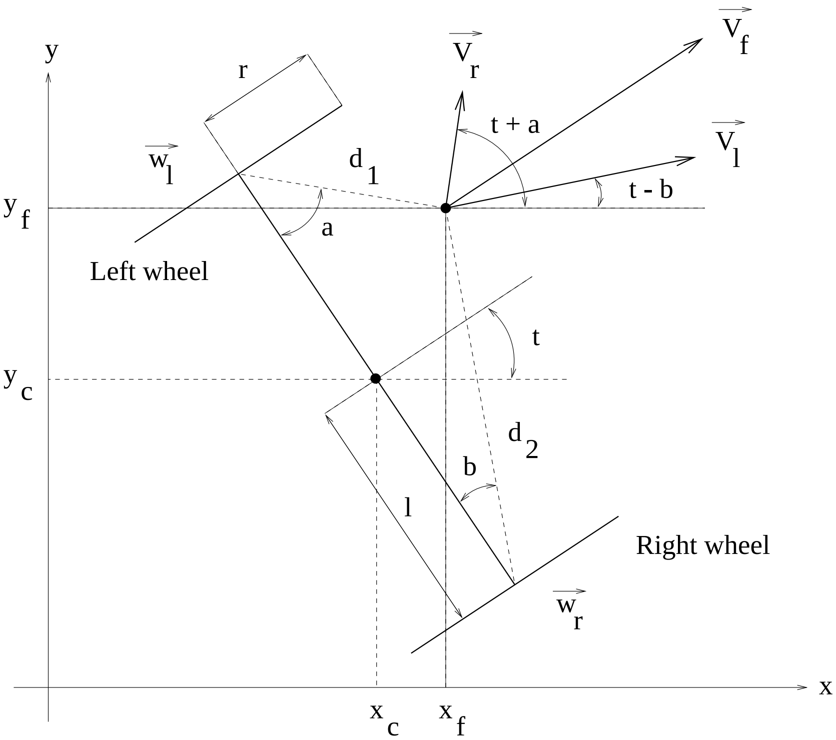



**Figure 7.** Illustration of the front-point idea for differential-drive mechanism. In the figure, the variables $\alpha$, $\beta$ and $\theta$ are represented by the letters a, b and t, respectively.

**Table 2.** Explanation of the terms used in deriving the reference point kinematics equations for differential-drive mechanism.

| **Term** | **Explanation** |
|---|---|
| $r$ | Radius of wheels |
| $l$ | Distance between the wheel center and the axle center |
| $d_1$ | Distance between left wheel center and the reference point |
| $d_2$ | Distance between right wheel center and the reference point |
| $\vec{w_r}$ | Angular velocity of right wheel (Magnitude = $w_r$) |
| $\vec{w_l}$ | Angular velocity of left wheel (Magnitude = $w_l$) |
| $\vec{V_r}$ | Linear velocity of the reference point due to right wheel (Magnitude = $V_r$) |
| $\vec{V_l}$ | Linear velocity of the reference point due to left wheel (Magnitude = $V_l$) |
| $\theta$ | Heading angle |
| $\alpha$ and $\beta$ | Angles (parameters) that determine the position of the reference point |

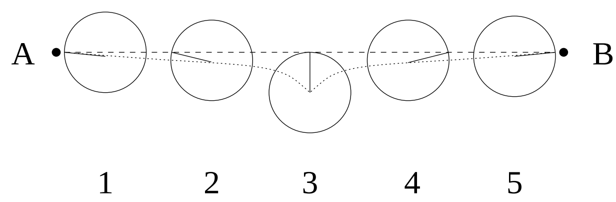



**Figure 8.** In this figure, a cylindrical synchro-drive robot starts its motion at point A (the front of the robot is facing Point A, but the robot's center is located below the line connecting the points A and B). The goal of the robot is to reach Point B. The figures numbered from 1 to 5 represent the robot configuration at the beginning of equal time intervals during the motion. The trajectories of the center and the front points are shown with dotted and dashed lines, respectively. Note that since the front point is holonomic, it can easily follow a straight path from Point A to Point B with either fixed or varying speed, but the center moves on a curved path in order to provide the front point with holonomic motion.

$$\begin{bmatrix} \dot{x_f} \\ \dot{y_f} \end{bmatrix} = \begin{bmatrix} rd_1 \cos(\theta+\alpha)/2l & rd_2 \cos(\theta-\beta)/2l \\ rd_1 \sin(\theta+\alpha)/2l & rd_2 \sin(\theta-\beta)/2l \end{bmatrix} \begin{bmatrix} w_r \\ w_l \end{bmatrix} = A \begin{bmatrix} w_r \\ w_l \end{bmatrix}.$$

The determinant of the matrix $A$ is:

$$\det(A) = -r^2 \sin\alpha \sin\beta / \sin(\alpha+\beta),$$

and is zero only when one or more of $\sin\alpha$ and $\sin\beta$ is zero (a logical assumption is; $r > 0$) which implies selecting the reference point on the main axis of the drive system (the main axis is the line that is aligned with the axle between the wheels). We can directly solve for $w_r$ and $w_l$ to obtain the inverse kinematics equation:

$$\begin{bmatrix} w_r \\ w_l \end{bmatrix} = A^{-1} \begin{bmatrix} \dot{x_f} \\ \dot{y_f} \end{bmatrix} = \begin{bmatrix} -\sin(\theta-\beta)/r\sin\beta & \cos(\theta-\beta)/r\sin\beta \\ \sin(\theta+\alpha)/r\sin\alpha & -\cos(\theta+\alpha)/r\sin\alpha \end{bmatrix} \begin{bmatrix} \dot{x_f} \\ \dot{y_f} \end{bmatrix}. \quad (2)$$

Given the heading angle $\theta$ and the desired velocity vector for the reference point $\vec{V_f} = \langle \dot{x_f}, \dot{y_f} \rangle$, a simple controller can use Equation (2) to find the correct angular velocities to be applied to the two wheels.

Computing the maximum speed attainable by the front point (or by any arbitrary reference point) and the constant maximum speed for the differential-drive case can also be done in the same way as we described in Sect. 5 for the synchronous-drive mechanism.

## 7. CONCLUSIONS

In this document, we re-examined the capabilities of some popular driving mechanisms from a different perspective, and showed that we can use them to provide holonomic motion to some qualifying reference points that we can substitute for the center of the driving mechanism (which is the usual choice as the reference in path-planning tasks).

By changing our reference point, we are also changing the regular type of motion exhibited by mobile robots. For example, Fig. 8 shows a situation that is not so uncommon when we employ the front-point method for robot navigation.

The method described in this document is a simple and very effective method that does not involve extensive mathematical computations and that can easily be implemented and applied on some popular driving mechanisms. For path-following tasks, our new reference points can achieve some seemingly difficult maneuvers easily and efficiently if we have the front-point method as a subprogram inside the controller for our robots.

## ACKNOWLEDGMENTS

This research is supported by the DARPA Mobile Autonomous Robot Software (MARS) program under contract DABT 63-99-1-0012.

## REFERENCES

1. J.-C. Latombe, *Robot Motion Planning*, ch. 9. Kluwer Academic Publishers, Boston, 1991.
2. G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*, Cambridge University Press, Cambridge, United Kingdom, 2000.
3. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research* **5**(1), pp. 90–98, 1986.
4. P. K. Agarwal, P. Raghavan, and H. Tamaki, "Motion planning for a steering-constrained robot through moderate obstacles," in *ACM Symposium on Theory of Computing*, pp. 343–352, 1995.
5. C. I. Connolly and R. A. Grupen, "Nonholonomic path planning using harmonic functions," Tech. Rep. 94-50, Computer Science Department, University of Massachusetts, May 1994.
6. R. C. Arkin, "Motor schema-based mobile robot navigation," *International Journal of Robotics Research* **8**, pp. 92–112, August 1989.
7. R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *IEEE International Conference on Robotics and Automation*, **4**, pp. 3375–3382, April 22–28, 1996.
8. D. Fox, W. Burgard, and S. Thrun, "Controlling synchro-drive robots with the dynamic window approach to collision avoidance," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.
9. K. Gupta and A. P. del Pobil, eds., *Practical Motion Planning in Robotics*, John Wiley, West Sussex, England, 1998.
10. T. Lozano-Pérez, "Spatial planning: A configuration space approach," *IEEE Transactions on Computers* **C-32**, pp. 108–120, February 1983.