

COPYRIGHT NOTICE



FedUni ResearchOnline

<http://researchonline.federation.edu.au>

This article is the version that was accepted for publication by Cambridge University Press. The final and definitive version of this article can be found at: <http://dx.doi.org/10.1017/S0263574710000317>

Percy, A., Spark, I. (2010) A numerical control algorithm for navigation of an operator-driven snake-like robot with 4WD-4WS segments. *Robotica*, 29(3), 471-482.

A numerical control algorithm for navigation of an
operator driven snake-like robot with 4WD-4WS
segments

Andrew Percy. Monash University, Australia.

Ian Spark. Gippsland Regional Automation Centre.

Yousef Ibrahim. Monash University, Australia.

Leon Hardy. University of South Florida, USA.

Abstract

This paper presents a new algorithm for the control of a snake-like robot with passive joints and active wheels. Each segment has four autonomously driven and steered wheels. The algorithm approximates the ideal solution in which all wheels on a segment have the same centre of curvature with wheel speeds providing cooperative redundancy. Each hitch point joining segments traverses the same path which is determined by an operator prescribing the

path curvature and front hitch speed. The numerical algorithm developed in this paper is simulation tested against a previously derived analytical solution for a predetermined path. Further simulations are carried out to show the effects of changing curvature and front hitch speed on hitch path, wheel angles and wheel speeds for a one, two and three segment robot.

KEYWORDS: snake-like robot, control system, 4WD-4WS, cooperative redundancy.

1 Introduction

In [1] an autonomous 4WD4WS operating system was introduced and this system was used in the design of a segmented gantry tractor in [2]. The design of the segmented gantry tractor is that of a snake-like robot consisting of segments each having four wheels controlled by the autonomous 4WD4WS operating system under the constraint that each hitch point, joining the segments, must follow the same path (see Figure 1). Snake-like robots of this type are called “passive bending joint and active wheel type” [3] although these robots generally have 2 wheels per segment with steering accomplished using only wheel speeds rather than wheel steering combined with speeds. Similar snake-like robots would also be suitable for segmented vehicles working in shaft mines, rescue robots carrying first aid supplies through small passages and other applications. With each segment having four wheels this snake-like

robot should be more stable under a load than those with two wheels per segment.

In [4] the analytic solutions for wheel angles and speeds of the first segment were determined for a gantry tractor traveling on a predetermined path through a gate. The algorithm for determining wheel angles and speeds would apply to any predetermined path which is given by spatial equations in a flat plane.

This paper gives a numerical algorithm to compute wheel angles and speeds approximating the analytical solutions for a path which is determined in real-time by an operator, either internal or external (for example [5]) to the robot. This involves determining the path traveled by the leading hitch of the first segment given the instantaneous input from the operator of curvature of the path, $\kappa(t)$, and velocity of the leading hitch-point, $v(t)$, at time t . The instantaneous radius of curvature, $\frac{1}{\kappa}$, is the radius of the circle best fitting the curved path locally to a given point. Negative κ indicates a right turn, $\kappa = 0$ indicates no turn and $\kappa > 0$ indicates a left turn. To determine the hitch path from these inputs, the algorithm of [6] was generalized by removing the restrictions of constant speed and “clothoid” curves, but in doing so, the property that arc length is proportional to time was lost, forcing the use of numerical techniques at an earlier stage of the path determination.

Once the path has been determined, knowing the length of each segment allows the position, on that path, of all hitch points to be found. There is then enough

information to calculate wheel angles and speeds in a number of ways. For high speed snake-like robots, issues of traction, yaw and scuffing would be significant and calculating the centre of gravity of each segment then using a kinematic, dynamic or hybrid system approach, perhaps with feedback [7, 8, 9, 10] could be more appropriate. The literature of these systems tends to neglect calculating time or use constant curvature path tracking, splines (for example [11]), or specific predetermined maneuvers such as lane changing. The common constraint that the front to rear, central axis be tangential to the path is also inappropriate for the snake-like constraint that all hitch points follow the same path. Alternatively, in [12], sensors have been proposed to enable a ‘platoon’ of vehicles to follow the same path.

Under the assumed conditions of a relatively slow moving robot on a relatively flat surface with no slip and negligible scuffing the geometrical control system given in [2] can be applied. This system simply calculates a common centre of curvature (COC) for the four wheels and two hitch points of each segment and the wheel speeds are given using ratios of radii from the COC. These wheel speeds provide a cooperative redundancy of two steering systems, with the advantage that when one system begins to fail it is backed up by the other system. A similar approach was taken in [13] but only the relation between wheel angle and turning radius was investigated there. The wheel angles and speeds can be given as functions of R_x and

R_y , the scalar components of the vector from the centre of segment (COS) to the COC, perpendicular and parallel to the inter-hitch axis of the segment [1]. R_x and R_y are shown in Figure 1.

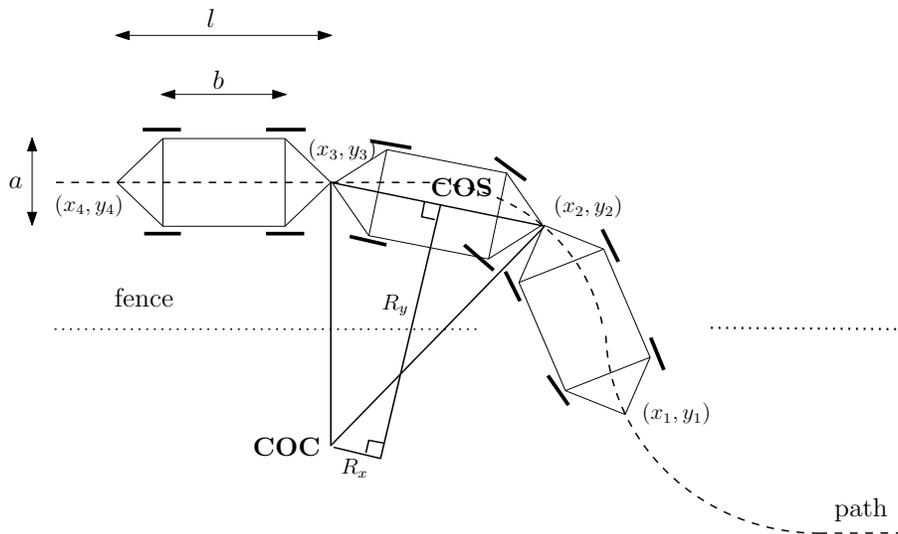


Figure 1: Dimensions, a , b and l , of the segments and components R_x and R_y of a snake-like robot passing through a gateway.

Since the calculation of wheel angles and speed takes a finite processing time, the proposed control system samples the operator's values of curvature and speed at a fixed time increment. The time increment must be sufficiently long to allow the calculation of the wheel angles and speeds of each segment before the control parameters are sampled again. The calculation of the path will involve the numerical approximation of several integrals. Trapezoidal approximation is used to facilitate

programmability and speed (similar to [14]). The simulation studies of Section 3 show that trapezoidal approximation is sufficiently accurate with a realizable time increment.

The dimensions of the segments, the range of speeds and of curvatures will effect the accuracy of the discrete, numerical calculations. If the path is too curved situations can develop where the rear hitch of a segment may equally well travel along the path in a forward or backward direction. This is avoided by restricting $\kappa < \frac{2}{l}$, with l the inter-hitch distance, so that the front and rear hitches cannot be at the ends of a diameter of a circle.

A simulation is carried out to demonstrate the accuracy of the numerical algorithm compared to an analytical solution, given in [4]. The analytical solution for a prescribed path was calculated with no attempt to remove singularities caused by the values of R_x or R_y approaching infinity under certain path conditions (for example, a straight path). The numerical algorithm was encoded in a *MATLAB* program for which it was necessary to algebraically modify the equations used in deriving the wheel speeds and angles, to ensure that all values remain finite and no singularities occur.

This paper concludes with several other simulations to validate the algorithm, using variable curvature, speed and number of segments. At this stage, resources are

not available for the construction of a snake-like robot with 4WD-4WS segments, and hence, no experimental testing of the proposed control algorithm was possible.

2 Numerical algorithm for wheel angles and speeds

2.1 Calculating the path

If the curvature of a path κ is given as a function of arc length, s , then using the Serret-Frenet equations for a planar path (one with no torsion) the path $\mathbf{x}(s)$ can be calculated as a vector function of arc length by the standard method of differential geometry [15]. Explicitly, the path is given by integrating the tangent vector $\dot{\mathbf{x}} = \frac{d}{ds}\mathbf{x}(s) = \mathbf{t}(s) = \cos(\theta(s))\mathbf{i} + \sin(\theta(s))\mathbf{j}$, for \mathbf{i} and \mathbf{j} the standard basis, where θ is the angle the path's tangent makes with the direction \mathbf{i} , calculated as $\theta(s) = \int_0^s \kappa(s) ds$. Figure 2 illustrates the vector components in the construction for calculating the path $\mathbf{x}(s)$.

For a real-time control system, it is preferable to work in the parameter of time rather than arc length. The standard method can be modified, introducing approximation techniques to determine discrete points along the path, $\mathbf{x}(t_i)$, as a function of time increments, $t_i = i\Delta t$, $i = 0, 1, 2, 3, \dots$, given its curvature $\kappa(t_i)$ and speed $v(t_i)$ at discrete time intervals. This discretized version of a path will be used to calculate

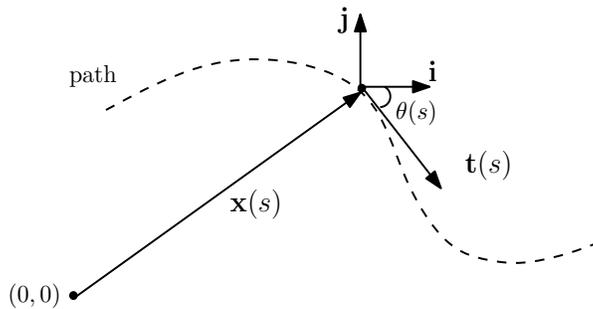


Figure 2: Vector construction for calculating the path $\mathbf{x}(s)$

discretized wheel angles and speeds for the segments of the snake-like robot using a method similar to that of [4]. The path of the front hitch will be given by the discrete points denoted $(x_1(t_i), y_1(t_i))$.

Initial conditions are necessary to approximate the definite integrals. Assume that initially $t_0 = 0$, $\kappa(t_0) = 0$, $v_1(t_0) = v_0$ m/s for some constant v_0 , distance traveled $s_1(t_0) = 0$, angle $\theta_1(t_0) = 0$, so the hitch is moving in the \mathbf{i} direction, and position $x_1(t_0) = 0$ and $y_1(t_0) = 0$.

The area of a (vertical) trapezoid is the average height times the width. For any $i > 0$, the i -th sample gives $\kappa(t_i)$ and $v_1(t_i)$ and the values $s_1(t_{i-1})$, $\theta_1(t_{i-1})$, $x_1(t_{i-1})$

and $y_1(t_{i-1})$ have previously been calculated. Then

$$\begin{aligned} s_1(t_i) &= s_1(t_{i-1}) + \frac{1}{2}(v_1(t_i) + v_1(t_{i-1}))(\Delta t) \\ \theta_1(t_i) &= \theta_1(t_{i-1}) + \frac{1}{2}(\kappa(t_i) + \kappa(t_{i-1}))(s(t_i) - s(t_{i-1})) \\ x_1(t_i) &= x_1(t_{i-1}) + \frac{1}{2}(v_1(t_i) \cos(\theta(t_i)) + v_1(t_{i-1}) \cos(\theta(t_{i-1}))) (\Delta t) \\ y_1(t_i) &= y_1(t_{i-1}) + \frac{1}{2}(v_1(t_i) \sin(\theta(t_i)) + v_1(t_{i-1}) \sin(\theta(t_{i-1}))) (\Delta t) \end{aligned}$$

can be calculated.

In this way, the discrete points $(x_1(t_i), y_1(t_i))$, $0 < i$ can be calculated in an iterative manner which is suitable for encoding into a loop program for machine calculation.

2.2 Calculation of the centre of curvature

Consider a snake-like robot with n segments. For simplicity of explication, assume each segment has an inter-hitch length of l meters. The algorithm can be modified to cater for segments of varying dimensions though setting up the initial conditions is more intricate. The j -th hitch will have coordinates labeled $(x_j(t), y_j(t))$ with speed $v_j(t)$, $1 \leq j \leq n + 1$.

All that is required to calculate the COC of the j -th segment is to determine the intersection of the normals to the path at the points of the j -th and $(j + 1)$ -th hitches. This requires first finding the position of the $(j + 1)$ -th hitch, which can be

done inductively if we know the position of the j -th hitch. The induction starts at the first hitch, the position of which was determined in Section 2.1.

2.2.1 Initial path conditions

Some initial conditions about the path are needed before the operator takes control. Assume that all segments are traveling straight at constant speed as the initial condition for all hitches. This condition is chosen for restarting the algorithm since errors in path calculation will increase constantly for straight sections and constant speed unless reset to zero, and they will remain zero until the next change of speed or curvature. Assume that Δt is sufficiently small that it is an integer divisor of l . It is necessary that the entire snake-like robot has traveled in a straight line at a constant speed v_0 for a distance of l so that the j -th hitch, $j = 1, 2, \dots, n$ has supplied a sequence of discrete path points for the $(j + 1)$ -th hitch to traverse.

Choose v_0 to be an integer divisor of $\frac{l}{\Delta t}$, so that all hitches will initially be positioned on discrete path points without error. That is, $qv_0\Delta t = l$ for some integer q . Then it is predetermined that the j -th hitch moves through the discrete points $(x_j(t_i), y_j(t_i)) = (-(j - 1)l + iv_0\Delta t, 0)$, $0 \leq i \leq q$ with $(-(j - 1)l + qv_0\Delta t, 0) = (-(j - 2)l, 0)$. For each hitch-point the associated speed is $v_j(t) = v_0$ and $\theta_j(t) = 0$ for $0 \leq i \leq q$. Thus, there will be a set of discrete path points of the j -th hitch for

the $(j + 1)$ -th hitch to follow for $i > q$ and $j > 1$.

2.2.2 The operator takes control

For time intervals $t_i > q\Delta t$, the position of each hitch is found inductively on j starting with a knowledge of the position of the first hitch as calculated in Section 2.1. Consider the position of the j -th hitch, at time $t = t_i$ with $i > q$ and assume that the position of the $(j - 1)$ -th hitch is known. Assume $(x_j(t_{i-1}), y_j(t_{i-1}))$ has been previously calculated in terms of the position of the $(j - 1)$ -th hitch at a previous time, say $(x_j(t_{i-1}), y_j(t_{i-1})) = (x_{j-1}(t_k), y_{j-1}(t_k))$, with $k \leq i - 1$.

The inter-hitch length of the $(j - 1)$ -th segment is l , so, if we have just calculated the point $(x_{j-1}(t_i), y_{j-1}(t_i))$ then we want to find the discrete point which is closest to a length l from $(x_{j-1}(t_i), y_{j-1}(t_i))$ to be the point $(x_j(t_i), y_j(t_i))$. We know all the discrete points for the $(j - 1)$ -th hitch up until time t_i . The lengths of some discrete points from $(x_{j-1}(t_i), y_{j-1}(t_i))$ are shown in Figure 3.

The discrete path point chosen for the j -th hitch at time t_i is the point $(x_{j-1}(t), y_{j-1}(t))$ that minimizes the value of $|(x_{j-1}(t_i) - x_{j-1}(t))^2 + (y_{j-1}(t_i) - y_{j-1}(t))^2 - l^2|$. The identity relies on the point closest away from $(x_{j-1}(t_i), y_{j-1}(t_i))$ having the square of its length closest to the value l^2 . The absolute value of this difference will always be positive so the point giving the minimum value will be the best approximation for

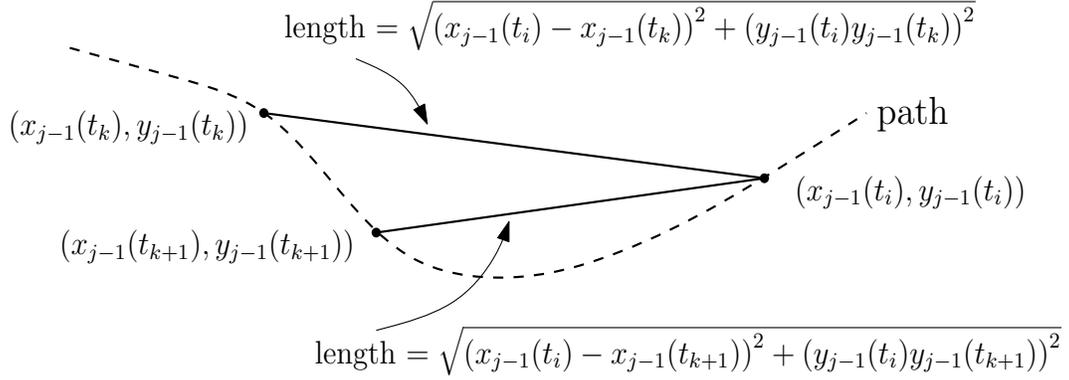


Figure 3: The length of discrete points $(x_{j-1}(t_k), y_{j-1}(t_k))$ and $(x_{j-1}(t_{k+1}), y_{j-1}(t_{k+1}))$ from the point $(x_{j-1}(t_i), y_{j-1}(t_i))$.

the discrete location of the j -th hitch at time t_i .

A calculation of this type for the point $(x_{j-1}(t), y_{j-1}(t))$ will be called a “Pythagorean test”. Assuming that the vehicle is always going forward along the path, or is stationary, then it is necessary to Pythagorean test the point $(x_j(t_{i-1}), y_j(t_{i-1})) = (x_{j-1}(t_k), y_{j-1}(t_k))$ and the points in front of this until a minimum value for the Pythagorean tests is found.

If the j -th hitch point $(x_j(t_i), y_j(t_i))$ is determined to be $(x_{j-1}(t_m), y_{j-1}(t_m))$ for $k \leq m < i$ then also set $\theta_j(t_i) = \theta_{j-1}(t_m)$.

2.2.3 Calculating the COC for the j -th segment at the i -th time increment

The tangent vector to the path of the j -th hitch at time t_i is given by $v_j(t_i) \cos(\theta_j(t_i)) \mathbf{i} + v_j(t_i) \sin(\theta_j(t_i)) \mathbf{j}$ and so a unit normal to the path, at the j -th hitch, is given by $\sin(\theta_j(t_i)) \mathbf{i} - \cos(\theta_j(t_i)) \mathbf{j}$. A parametric equation of the line normal to the path at the j -th hitch (at the front of the j -th segment), at time t_i , is given, as a function of the parameter $g_{j,j}$, by

$$\mathbf{N}_{\mathbf{j},\mathbf{j}}(g_{j,j}) = (x_j(t_i) + g_{j,j} \sin(\theta_j(t_i))) \mathbf{i} + (y_j(t_i) - g_{j,j} \cos(\theta_j(t_i))) \mathbf{j} \quad (1)$$

This vector function traces out the position vector of the points on the normal line for varying values of length $g_{j,j}$ from $(x_j(t_i), y_j(t_i))$, as shown in Figure 4.

Similarly, a parametric equation of the line normal to the path at the $(j+1)$ -th hitch (at the rear of the j -th segment) is given by

$$\mathbf{N}_{\mathbf{j}+1,\mathbf{j}}(g_{j+1,j}) = (x_{j+1}(t_i) + g_{j+1,j} \sin(\theta_{j+1}(t_i))) \mathbf{i} + (y_{j+1}(t_i) - g_{j+1,j} \cos(\theta_{j+1}(t_i))) \mathbf{j} \quad (2)$$

Equations (1) and (2) can be solved simultaneously to find the values of $g_{j,j}$ and $g_{j+1,j}$ making $\mathbf{N}_{\mathbf{j},\mathbf{j}}(g_{j,j}) = \mathbf{N}_{\mathbf{j}+1,\mathbf{j}}(g_{j+1,j})$, where the two normal lines meet at the COC.

The values of $|g_{j,j}(t_i)|$ and $|g_{j+1,j}(t_i)|$ will also give the distance of the COC from the j -th and $(j+1)$ -th hitches respectively since unit normal vectors were used in

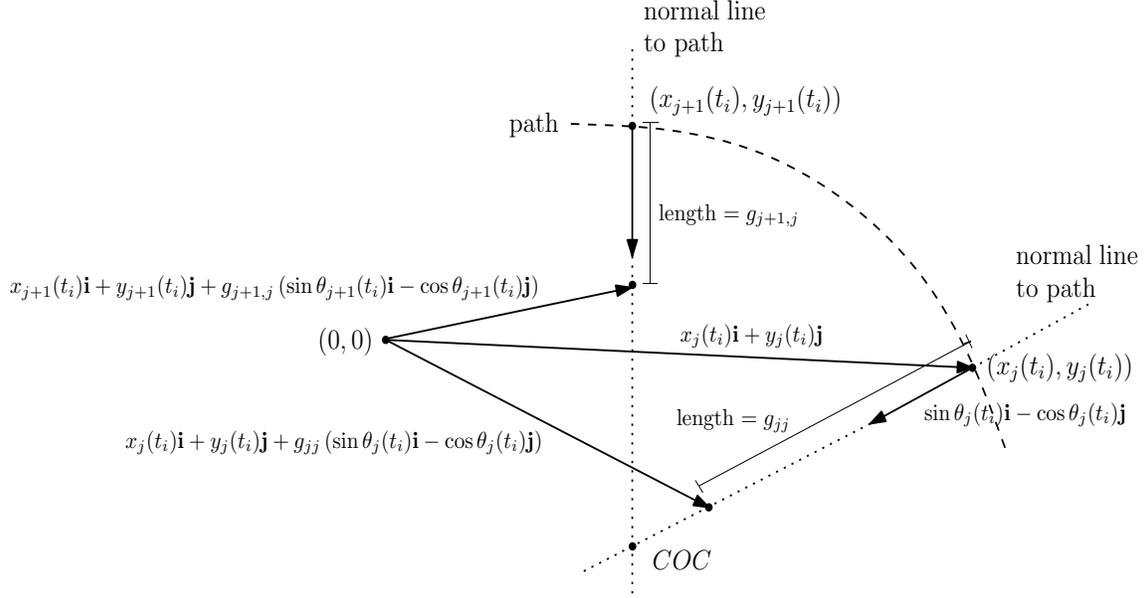


Figure 4: Defining the lines $\mathbf{N}_{j,j}(\mathbf{g}_{j,j})$ and $\mathbf{N}_{j+1,j}(\mathbf{g}_{j+1,j})$ as functions of length from the j -th and $(j + 1)$ -th hitch points.

the parametric equations for the normal lines (see Figure 4). Consequently, if $v_j(t_i)$ is the speed of the j -th hitch at time $t = t_i$ then $\frac{|g_{j,j}(t_i)|}{|g_{j+1,j}(t_i)|} = \frac{v_j(t_i)}{v_{j+1}(t_i)}$ allowing the calculation of (noting the sign of $g_{j,j}(t_i)$ and $g_{j+1,j}(t_i)$ must be the same so absolute values are unnecessary)

$$v_{j+1}(t_i) = \frac{g_{j+1,j}(t_i)v_j(t_i)}{g_{j,j}(t_i)} \quad (3)$$

which is the speed of the $(j + 1)$ -th hitch. Since the speed of the first hitch is known, all hitch speeds can be calculated inductively.

It is intuitively clear that the values of $g_{j,j}(t_i)$ and $g_{j+1,j}(t_i)$ will become infinite as

the two normal lines approach parallel. In order to avoid these undefined solutions the subsequent calculations will be manipulated using the fact that, since $\kappa(t) < \frac{l}{2}$, the COC can never be located at a hitch point and consequently, the values $G_{j+1,j}(t_i) = \frac{1}{g_{j+1,j}(t_i)}$ and $G_{j,j}(t_i) = \frac{1}{g_{j,j}(t_i)}$ are always finite.

2.3 Overcoming singularities in the control of wheel speeds and angles

The wheel angles and speeds for each module, at time t_i , are calculated during the time increment Δt . All quantities used in the calculations are evaluated for time t_i so, to simplify notation, the reference to time is suppressed. During Section 2.3, at time t_i , denote hitch points as (x_j, y_j) , angles θ_j , hitch speeds v_j and the parameter associating the $(j + 1)$ -th hitch to the COC of the j -th segment by $G_{j+1,j}$.

Solving $\mathbf{N}_{j,j}(g_{j,j}) = \mathbf{N}_{j+1,j}(g_{j+1,j})$ leads to

$$G_{j+1,j} = \frac{-\sin(\theta_{j+1} - \theta_j)}{(\sin(\theta_j))(y_{j+1} - y_j) + (\cos(\theta_j))(x_{j+1} - x_j)} \quad (4)$$

and

$$G_{j,j} = \frac{-\sin(\theta_{j+1} - \theta_j)}{(\sin(\theta_{j+1}))(y_{j+1} - y_j) + (\cos(\theta_{j+1}))(x_{j+1} - x_j)} \quad (5)$$

As stated, these are both finite quantities. This can be seen since, for the denominator to equal 0 would require $\tan(\theta_{j+1}) = \frac{-(x_{j+1} - x_j)}{(y_{j+1} - y_j)}$, which states that the

tangent to the path at the $(j + 1)$ -th hitch is perpendicular to the inter-hitch axis of the j -th segment which violates the $\kappa(t) < \frac{l}{2}$ restriction.

The planar coordinates of the COC of the j -th segment are multiplied by $G_{j+1,j}$ to get scaled coordinates:

$$(C_{j,1}, C_{j,2}) = (G_{j+1,j}(x_{j+1}) + \sin \theta_{j+1}, G_{j+1,j}(y_{j+1}) - \cos \theta_{j+1})$$

The coordinates of the COS of the j -th segment are also multiplied by $G_{j+1,j}$ to get scaled coordinates:

$$(M_{j,1}, M_{j,2}) = \left(\frac{G_{j+1,j}}{2}(x_{j+1} + x_j), \frac{G_{j+1,j}}{2}(y_{j+1} + y_j) \right)$$

Then the vector from the j -th segment's COS to its COC, multiplied by $G_{j+1,j}$, is $\mathbf{R}_j = (C_{j,1} - M_{j,1})\mathbf{i} + (C_{j,2} - M_{j,2})\mathbf{j}$. A vector parallel to the axis of the j -th segment from the $(j + 1)$ -th hitch to the j -th hitch is $\mathbf{L}_j = (x_j - x_{j+1})\mathbf{i} + (y_j - y_{j+1})\mathbf{j}$ and consequently a vector perpendicular to this axis (suitably oriented) is $\mathbf{P}_j = (y_j - y_{j+1})\mathbf{i} - (x_j - x_{j+1})\mathbf{j}$.

Resolving \mathbf{R}_j into its components in the directions of \mathbf{L}_j and \mathbf{P}_j by the standard 'dot' product method gives the desired R_x and R_y multiplied by a factor of $G_{j+1,j}$, which we denote

$$T_{j,x} = \frac{(C_{j,1} - M_{j,1})(y_j - y_{j+1}) - (C_{j,2} - M_{j,2})(x_j - x_{j+1})}{\sqrt{(x_j - x_{j+1})^2 + (y_j - y_{j+1})^2}} \quad (6)$$

$$T_{j,y} = \frac{(C_{j,1} - M_{j,1})(x_j - x_{j+1}) + (C_{j,2} - M_{j,2})(y_j - y_{j+1})}{\sqrt{(x_j - x_{j+1})^2 + (y_j - y_{j+1})^2}} \quad (7)$$

The wheel angles of a segment are calculated by trigonometry in [1] and, accounting for the multiplicative factor of $G_{j+1,j}$, are given, for the j -th segment at time t_i , in Table 1. Positive wheel angles are clockwise for the front and anticlockwise for the rear wheels.

Wheel speeds are calculated using the fact that the ratio of wheel hub distance, from the COC, to hitch distance, from the COC, is equal to the ratio of the translational hub speed to hitch speed, since both wheel and hitch are revolving around the same centre of curvature. For the j -th segment, the speed of the j -th hitch, at time t_i , was determined using equation (3) as $v_j = \frac{G_{j-1,j-1}v_{j-1}}{G_{j,j-1}}$, and the length of the front hitch from the COC is $|g_{j,j}| = \left| \frac{1}{G_{j,j}} \right|$. The length of the wheel hub from the COC is calculated by trigonometry in [1] and consequently, accounting for the multiplicative factor of $G_{j+1,j}(t_i)$, wheel speeds for the j -th segment, at time t_i , are given with the corresponding wheel angle in Table 1.

Both the functions for wheel angle and wheel speeds are given in terms of the values of $G_{j,j}$, $G_{j+1,j}$, $T_{j,x}$ and $T_{j,y}$. Each of these values, and hence the wheel angles and speeds, are completely determined from the two input variables of κ and v . As a result, the wheel angles and speeds given in Table 1, that ensure all hitch points remain on the path of the front hitch, are determined as functions of the operator's prescribed values of $\kappa(t_i)$ and $v(t_i)$.

Wheel	Wheel angle	Wheel speed
Right Front	$\arctan\left(\frac{bG_{j+1,j} - 2T_{j,y}}{2T_{j,x} - aG_{j+1,j}}\right)$	$\frac{v_j G_{j,j}}{2G_{j+1,j}} \sqrt{(bG_{j+1,j} - 2T_{j,y})^2 + (2T_{j,x} - aG_{j+1,j})^2}$
Left Front	$\arctan\left(\frac{bG_{j+1,j} - 2T_{j,y}}{2T_{j,x} + aG_{j+1,j}}\right)$	$\frac{v_j G_{j,j}}{2G_{j+1,j}} \sqrt{(bG_{j+1,j} - 2T_{j,y})^2 + (2T_{j,x} + aG_{j+1,j})^2}$
Right Rear	$\arctan\left(\frac{bG_{j+1,j} + 2T_{j,y}}{2T_{j,x} - aG_{j+1,j}}\right)$	$\frac{v_j G_{j,j}}{2G_{j+1,j}} \sqrt{(bG_{j+1,j} + 2T_{j,y})^2 + (2T_{j,x} - aG_{j+1,j})^2}$
Left Rear	$\arctan\left(\frac{bG_{j+1,j} + 2T_{j,y}}{2T_{j,x} + aG_{j+1,j}}\right)$	$\frac{v_j G_{j,j}}{2G_{j+1,j}} \sqrt{(bG_{j+1,j} + 2T_{j,y})^2 + (2T_{j,x} + aG_{j+1,j})^2}$

Table 1: Wheel angles and speeds of the j -th segment at time t_i

The fact that $G_{j,j}$ and $G_{j+1,j}$ have the same sign has again been used to remove the need for absolute values to be taken. Wheel speed is given as the translational speed of the hub in units of distance per second. For a wheel of radius r units of distance, the rotational speed of any of the wheels in radians per second is given by dividing the translational speed by r .

The value of $G_{j+1,j}$ can be zero, however, as discussed for equations (4) and (5), under the restriction $\kappa(t) < \frac{l}{2}$, the ratio

$$\frac{G_{j,j}}{G_{j+1,j}} = \frac{(\sin(\theta_j))(y_{j+1} - y_j) + (\cos(\theta_j))(x_{j+1} - x_j)}{(\sin(\theta_{j+1}))(y_{j+1} - y_j) + (\cos(\theta_{j+1}))(x_{j+1} - x_j)}$$

has no singularities so that no singularities occur when calculating wheel speeds.

3 Simulations and discussion

In order to validate the developed control algorithm, simulation studies were carried out for various operating conditions. The numerical algorithm is coded into a *MATLAB* program which, for each value of $t_i = i\Delta t$, samples values of $\kappa(t_i)$ and $v_1(t_i)$ from external files, simulating the sampling of the operator's input. That is, the operator is also a simulation given as functions for $\kappa(t)$ and $v(t)$, external to the main algorithm program. The program then calculates the hitch points and the segment wheel angles and speeds for t_i before sampling the values from the external files for $\kappa(t_{i+1})$ and $v_1(t_{i+1})$ and recalculating the segment wheel angles and speeds for t_{i+1} . The loop continues for counting numbers i up to a predetermined value.

3.1 Accuracy testing of the numerical algorithm

To test the accuracy of the numerical algorithm the current calculations of wheel speeds and angles will be compared, using simulation, with those calculated for a predetermined path in [4]. The analytic solutions in [4] are for a single segment with the segment dimensions shown in Figure 1 of $a = 0.5$ m, $b = 0.5$ m and $l = 1$ m. The operator's parameters for the test simulation are a constant hitch speed $v_1 = 1$

m/s and curvature,

$$\kappa(t) = \begin{cases} 0 & 0 < t \leq 1 \\ -1 & 1 \leq t \leq 1 + \pi/2 \\ 1 & 1 + \pi/2 \leq t \leq 1 + \pi \\ 0 & 1 + \pi \leq t \leq 3 + \pi \end{cases} \quad (8)$$

This gives the path of a scaled model of a gantry tractor, with 5 m long segments, passing through a gate, with wheel speeds around 18 km/h, using straight paths and circular arcs as shown in Figure 1. The assumption of no slip should remain valid in such a case given hard ground and good wheel traction, since there is a co-operative redundancy between the steering systems of wheel angle and wheel speed. Different conditions such as muddy ground would not satisfy the non-slip assumption.

The accuracy of the algorithm depends on the value of Δt . Decreasing Δt increases the number of discrete points along the path. Since, with smaller Δt the discrete points are closer together, there will be less error in the choice of a discrete point approximating the true length of the module. However, Δt must be sufficiently large to allow for the calculation of wheel speeds and angles before the next sampling occurs in the recursive process. The most natural choice for a value of Δt is the smallest value with sufficient margin of error to allow certainty of the calculation.

The *MATLAB* simulation was run 10 times (on a desktop PC using *Windows XP*

operating system, 512 MB RAM, processor speed 2593 MHz) with each of the Δt values shown in Table 2. The results indicate that choosing $\Delta t = 0.001$ seconds will give a suitable balance of calculating speed and accuracy. We will use $\Delta t = 0.001$ s for all simulations of Section 3.1. In the simulation graphs, the horizontal scale is in increments of 0.001 s. For example, 4142 increments of 0.001 s represents a time of 4.142 s, which is the closest time increment to $t = 1 + \pi$ seconds.

Δt (seconds)	mean (seconds)	standard deviation (seconds)	mean relative to Δt
0.01	0.000219	0.000044	0.02 Δt
0.001	0.000659	0.0000004	0.66 Δt
0.0005	0.001229	0.0000002	2.46 Δt

Table 2: CPU time per program loop for simulation with different Δt

There are two sources of error in the algorithm; error by approximating the integrals to calculate the discrete path points and error in selecting a discrete path point to represent the position of the rear hitch point.

The accuracy of determining the discrete path points using the numerical algorithm is shown in Figure 5. The error shown is the distance of the discrete point away from where the hitch should be in the analytical solution of [4]. The mean

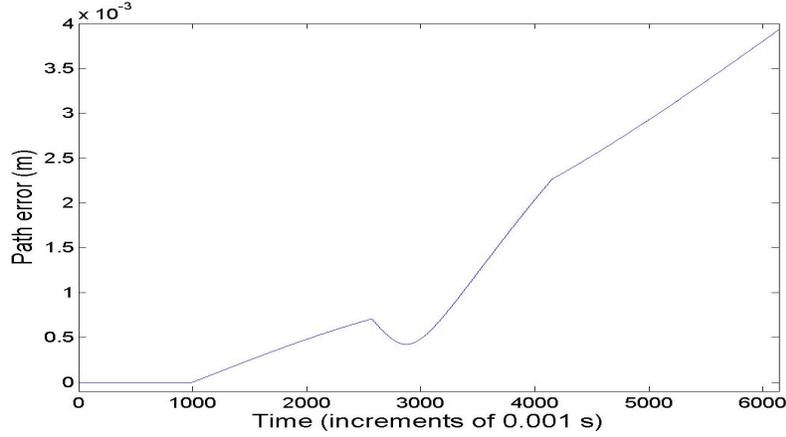


Figure 5: Distance between numerical path point and actual path point with $\Delta t = 0.001$

path error is 1.4×10^{-3} m with standard deviation 1.3×10^{-3} m, for this particular simulation. The rate of change of the error is not smooth at the discontinuous points in the function for κ , as would be expected. The continuing increase in error, after $t = 1 + \pi$, when the hitch should return to a straight line in the direction \mathbf{i} , is due to the fact that $\theta(t_i)$ does not return exactly to zero before becoming constant whilst $\kappa = 0$, after $t = 1 + \pi$. This has a small effect in the calculation of $y_1(t_i)$ since $\sin \theta$ has a slope close to 1 near zero. The effect on $x_1(t_i)$ is negligible since $\cos \theta$ has a slope close to 0 near zero. This highlights the need to restart the algorithm when the front hitch is in a straight path but this would require a directional sensor to determine the true value of θ .

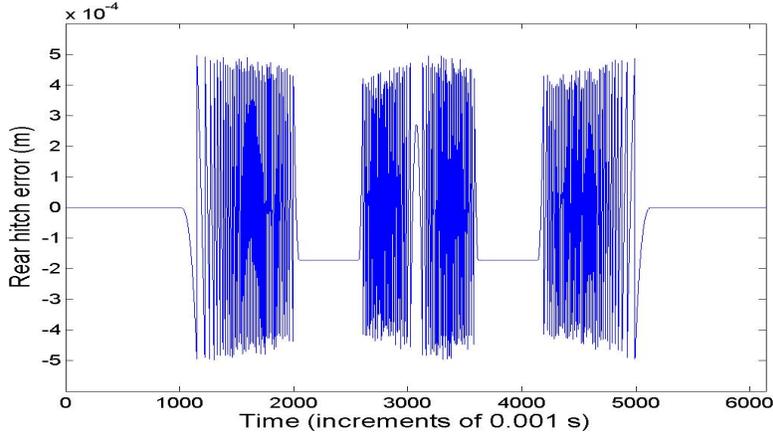


Figure 6: Error in distance between front and rear hitches with $\Delta t = 0.001$

The accuracy of determining the position of the rear hitch as a discrete point compared to the position determined analytically in continuous time is demonstrated in Figure 6 which graphs the difference between inter-hitch distance determined numerically and the true value of 1. We see a maximum error of 5×10^{-4} m, as expected. The mean inter-hitch error is -3.2×10^{-5} m with standard deviation 2.0×10^{-4} m, for this particular simulation. The “jittery” effect in which we observe ‘vertical sections’ on the error plot occurs due to sudden changes in the sign of the error. These sudden changes correspond to a change in choice from the $(i - n)$ -th to the $(i - (n \pm 1))$ -th discrete point with $n \in \{100, 99, 98\}$. For instance, when traveling along the initial straight section the rear hitch will be at discrete point $(x_2(t_i), y_2(t_i)) = (x_1(t_{i-100}), y_1(t_{i-100}))$ one metre behind the front hitch. However,

as the front hitch enters the first curve, the discrete point $(x_1(t_{i-100}), y_1(t_{i-100}))$ slowly becomes smaller than the length of the module, $l = 1$ metre, resulting in an increasing, negative error until suddenly, at one particular time increment, the discrete point $(x_1(t_{i-101}), y_1(t_{i-101}))$ is Pythagorean tested to be the closest discrete point to being 1 metre from the front hitch and is selected as the position for the rear hitch, now with positive error. We observe no “jitter” when both front and rear hitch are on the same straight section or curved section where the constant equal curvature of the path at front and rear hitch creates a “steady state”.

Figure 7 compares front hitch path, wheel angles and speeds calculated analytically with those of the numerical solution, with all four wheel angles or speeds on one graph for each solution. In each legend RF denotes the right front wheel values, LF denotes the left front, RR the right rear and LR the left rear wheel values. We observe from the plots for wheel angles the expected symmetry of diagonally opposed wheels for time in opposite directions. The negative values for the angle of the rear wheels just after the front hitch reaches the first circular arc (at $t=1$ s) are due to the fact that the hitch point is behind the rear hubs which must move to their right as the front hitch moves around the circular arc to the right. No symmetry of forward and reverse direction occurs for wheel speeds since it is a condition that the front hitch has constant speed along the path whereas the rear hitch has a variable speed.

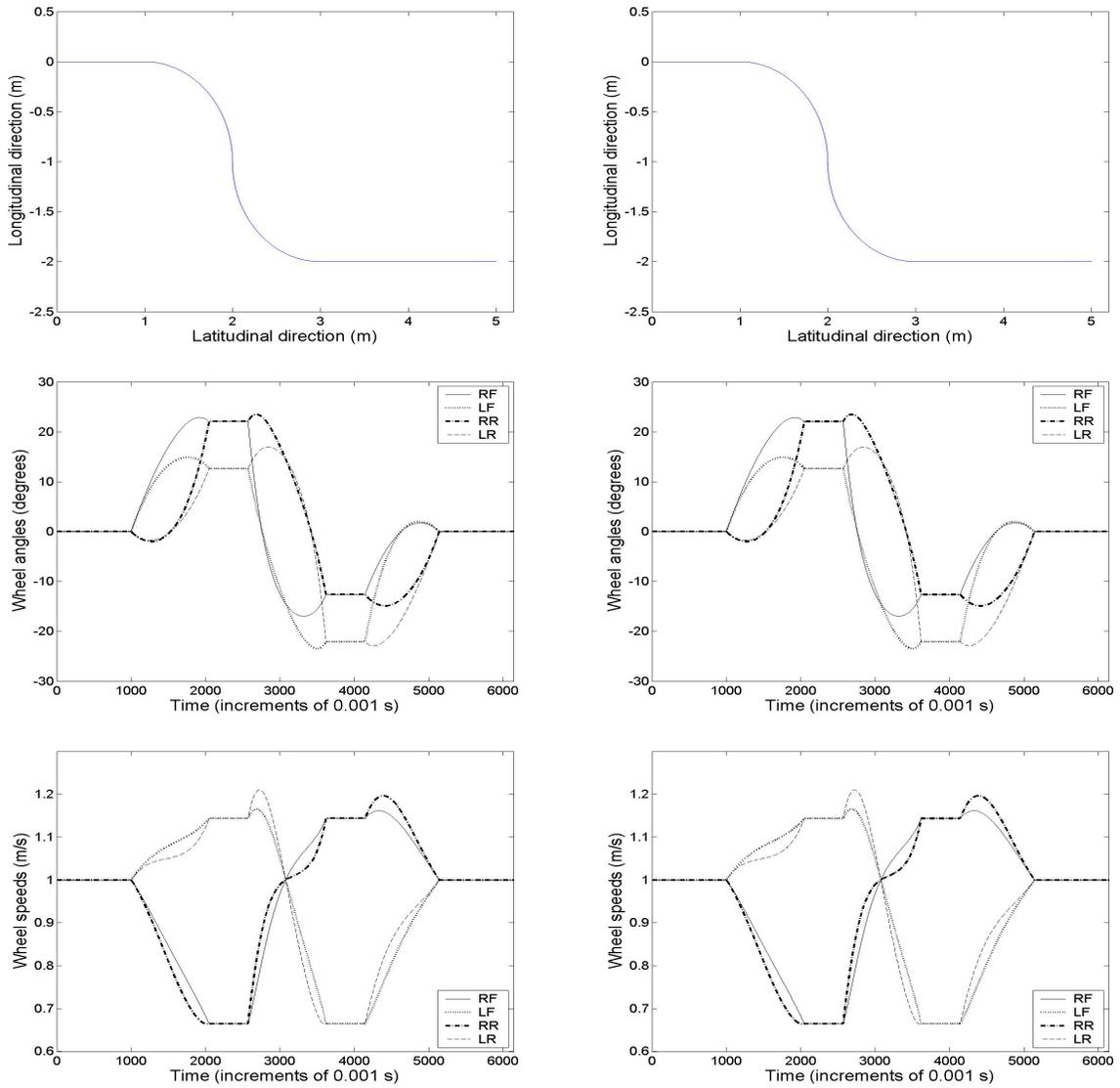


Figure 7: Comparison of numerical and analytical calculations. The left column shows the analytical front hitch path, wheel angles and wheel speeds. The right column shows the numerical front hitch path, wheel angles and wheel speeds.

The slowing of inner wheels and speeding up of outer wheels occurs as expected around the circular arcs. The steady states, between transitions, show left and right side wheels having the same constant values for both angles and speeds, as expected when in these states.

To visualize the error, the difference of analytical and numerical values of wheel angles and speeds are plotted for each wheel in Figure 8. Combining all wheels the calculated mean angle error is 4.9×10^{-5} degrees with a standard deviation of 1.6×10^{-4} degrees. Similarly, the mean speed error is -1.8×10^{-5} m/s with a standard deviation of 9.3×10^{-5} m/s. Again there is a “jittery” effect with sudden changes in the sign of the error. This follows from the change in error of the position of the rear hitch which is an essential component in the calculation of wheel speeds and errors. Again we see no jitter when the module is in a “steady state” condition.

3.2 Simulation study with a three segment robot

The average errors for both wheel angles and speeds are relatively small, as are the standard deviations, so we can conclude that the numerical algorithm gives an accurate approximation of wheel angles and speeds over the entire maneuver. Satisfied that the numerical algorithm was sufficiently accurate it was used to calculate wheel angles and speeds of the second and third segments. The algorithm defines the front

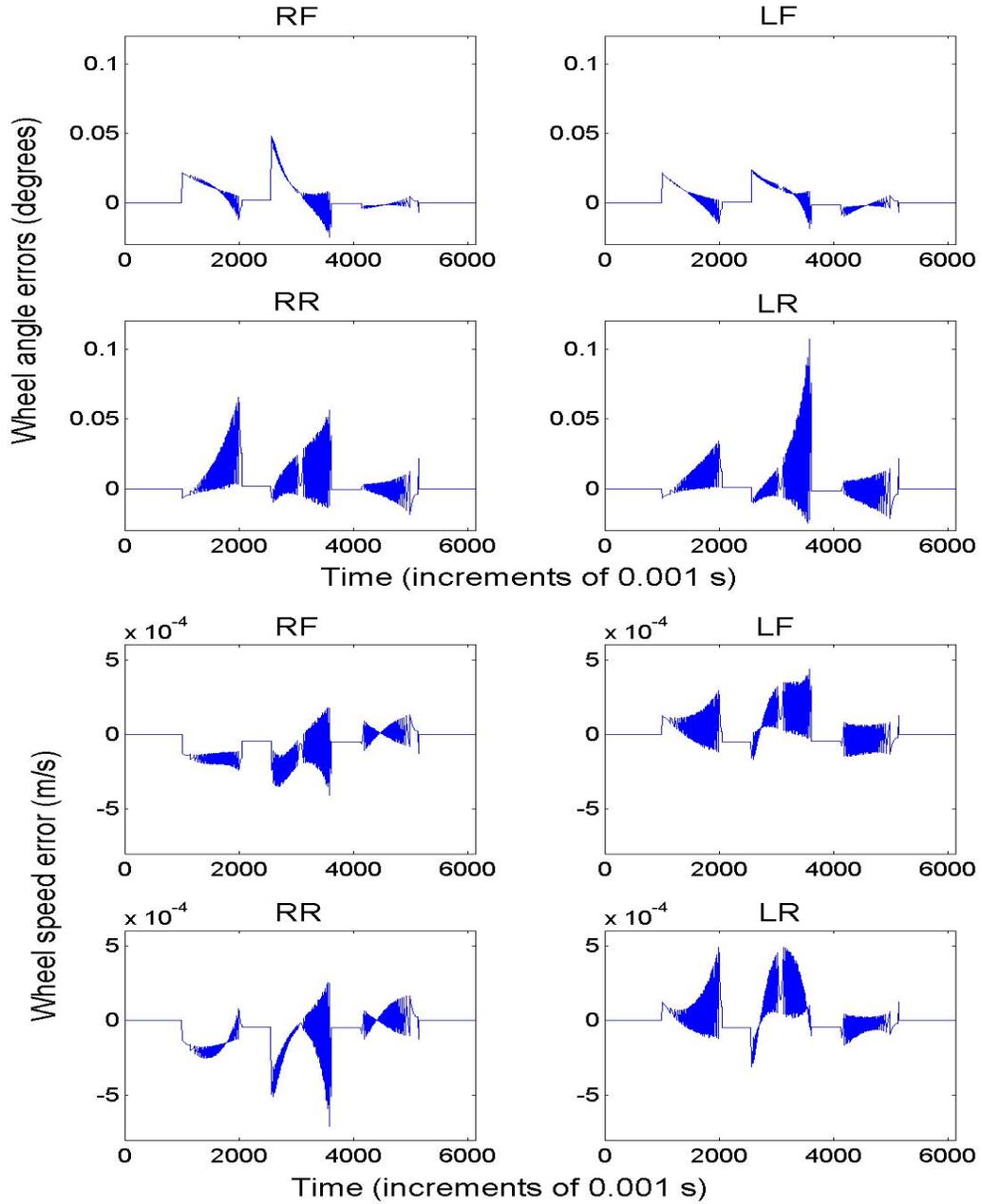


Figure 8: Wheel angle errors for individual wheels with $\Delta t = 0.001$ in upper four graphs and speed errors in lower four graphs

hitch of the second module to be at exactly the same discrete point as the rear hitch of the first module, so in theory there is no need for a “hitch pin” joining the modules. In an experiment the error due to discretization of the path will cause some forces on a real “hitch pin” but since the errors are low these forces would have negligible effect on the assumption of no wheel slip.

The κ of equation (8) is slightly modified by extending the last period of zero curvature to be $\kappa(t) = 0, 1 + \pi \leq t \leq 4 + \pi$. This extends the last straight section of the path, which is otherwise identical to that in row 1, column 2 of Figure 7, by one metre, ensuring all modules finish the simulation on a straight section of path. Using this modified κ and constant front-hitch speed $v_1 = 1$ m/s, the wheel angles and speeds for the first three segments are calculated in a single loop of the *MATLAB* program. The additional calculation for the second and third segments increases the calculation time per loop. For 10 runs with $\Delta t = 0.002$ s the average CPU time per loop was $1.14 \times 10^{-3} = 0.57\Delta t$. Figure 9 shows the results for wheel angles and speeds of the first segment in the first row, the second segment in the second row and the third segment in the third row.

Because the average wheel speeds stay relatively close to 1 m/s the periods of angle transition appear identical for each segment, although translated due to the later time subsequent segments enter the transition states. As expected, all wheels

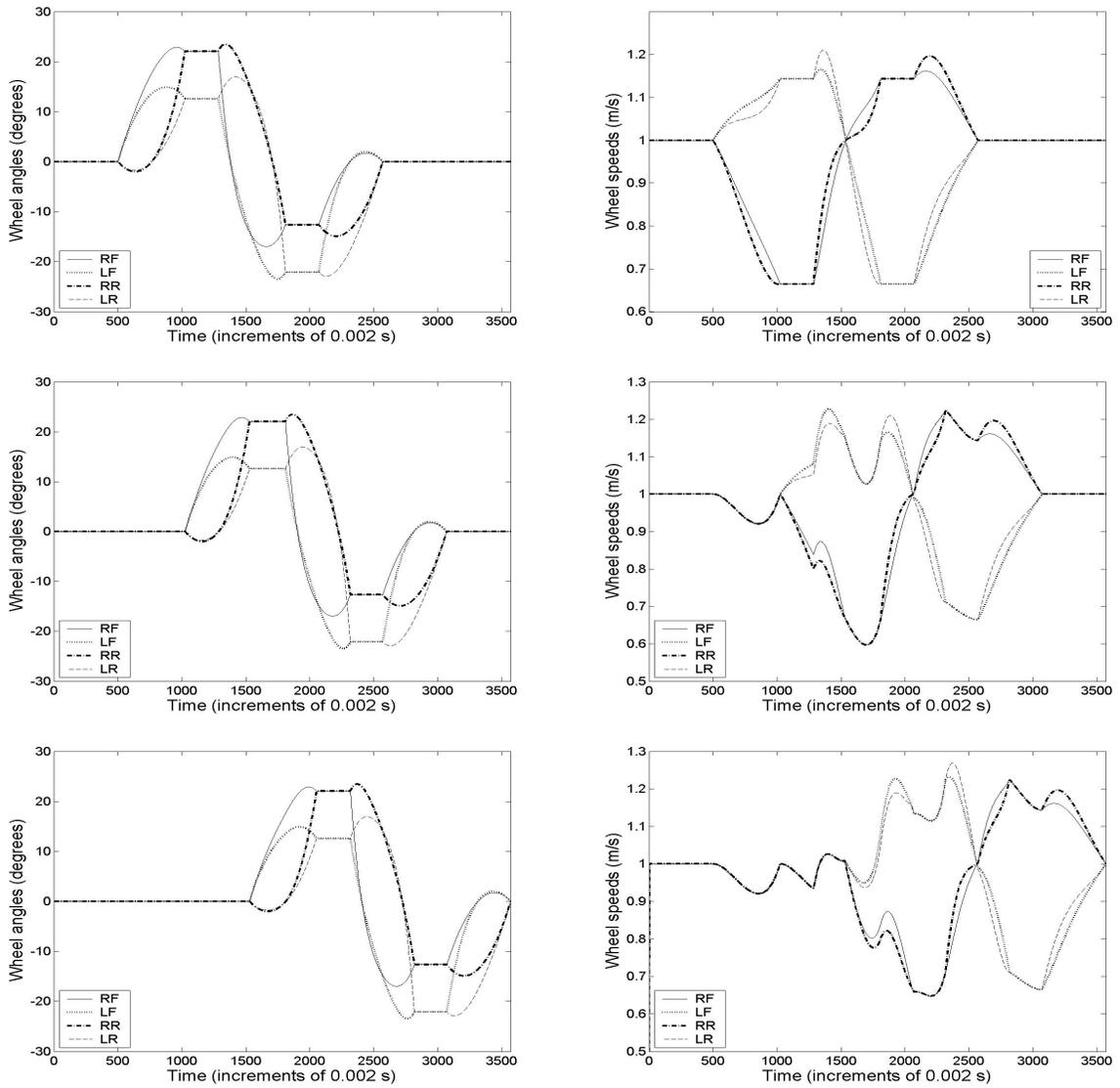


Figure 9: Wheel angles and speeds for the first segment in row 1, the second segment in row 2 and the third in row 3

show a change of speed as soon as the front hitch reaches the first circular arc and, because the later segments are traveling on a straight path, all their wheel speeds are the same until angle transition occurs. All wheels of each segment also have the same value of 1 m/s when the segment is half way through the transition between circular arcs.

3.3 Further simulation with one segment

Two further simulations were carried out for one segment with $\Delta t = 0.001$ s. The left column of Figure 10 shows the hitch path, wheel angles and speeds, using κ similar to that given in equation (8),

$$\kappa(t) = \begin{cases} 0 & 0 < t \leq 2 \\ -1 & 2 \leq t \leq 2 + \pi/2 \\ 1 & 2 + \pi/2 \leq t \leq 2 + \pi \\ 0 & 2 + \pi \leq t \leq 3 + \pi \end{cases} \quad (9)$$

but varying velocity linearly as a function of time, $v(t) = 0.2 + 0.3(t)$. The initial time interval that the front hitch is moving in a straight line in equation (9) has been extended so that, with the reduced speeds, it still moves forward through the length of one segment in a straight line, thereby prescribing a complete set of points for the second hitch to follow. A similar approach will allow for a standing start.

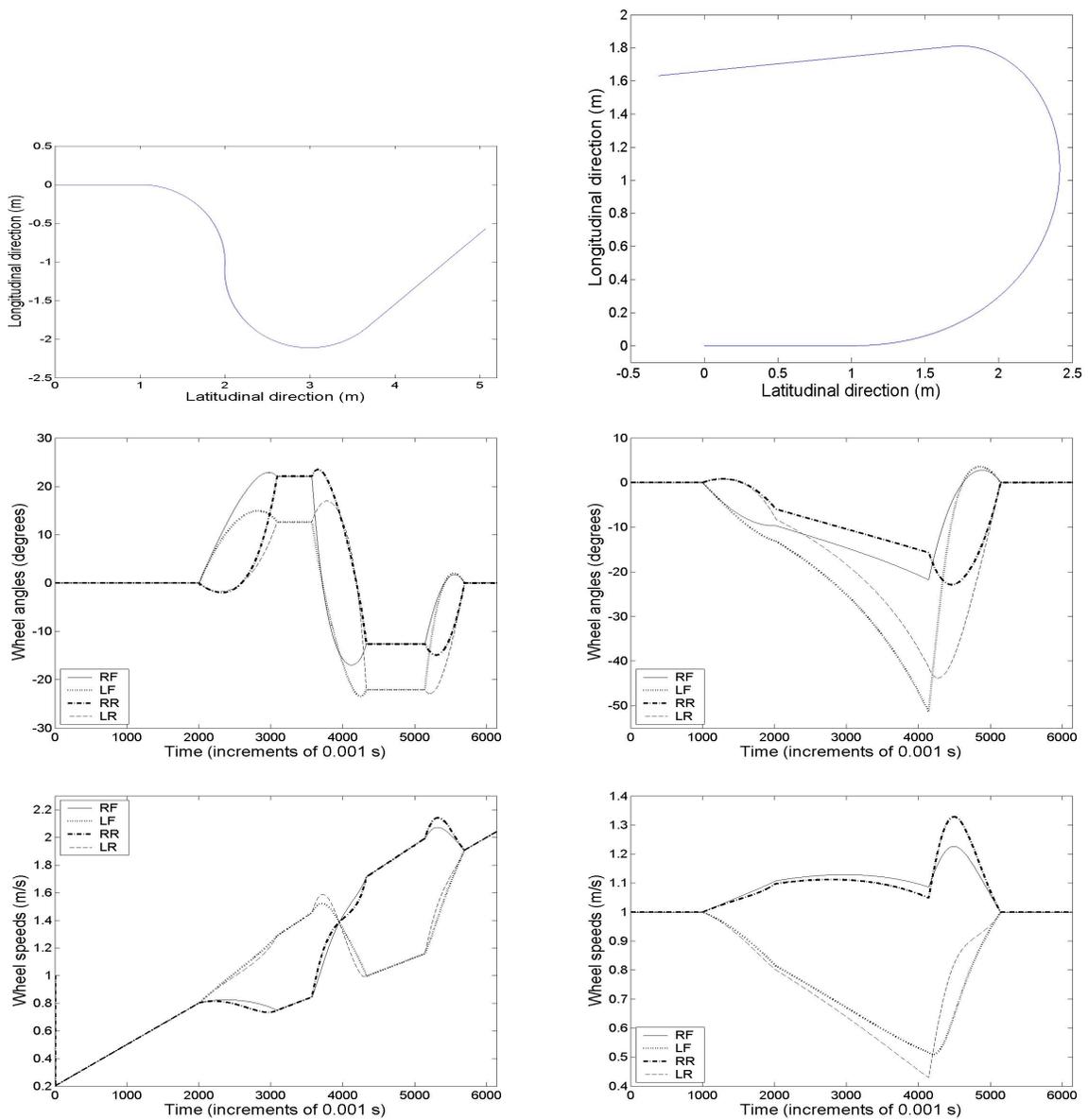


Figure 10: Further simulation showing path, wheel angle and wheel speed with varying speed in the left column and varying curvature in the right column

The shape of the graphs are as expected. The average speed for the first straight section is 0.5 m/s so the straight path of the front hitch is 1 m long. Over the next $\frac{\pi}{2}$ s the average speed is 1.0 m/s so a quarter right circle is covered. The average speed for the next $\frac{\pi}{2}$ s is 1.5 m/s so $\frac{3}{8}$ of a left circular arc is covered and the average speed for the last 1 s is 1.9 m/s so a 1.9 m straight section finishes the path. The graph of wheel angles shows a time distorted replica of the wheel angles in Figure 7, with transitions occurring increasingly quickly and steady states being extended as speed increases. The graph of wheel speeds shows a time sheared replica of the wheel speeds in Figure 7, with transitions occurring increasingly quickly and steady states being replaced by linearly increasing sections which are extending in duration as speed increases.

The right column shows hitch path, wheel angles and speeds with constant speed $v_1(t) = 1$ m/s and a linearly increasing curvature $\kappa(t) = 0.4(t)$ for $1 < t \leq 1 + \pi$ and $\kappa(t) = 0$ otherwise.

The shape of the graphs are as expected. The front hitch path shows a straight section of 1 m covered in the first second followed by a clothoid left turn over the next π s completed with a 2 m straight section covered over the last 2 s. The wheel angles show a similarity to the first transition of Figure 7 but instead of a steady state following the initial transition, the angles continue to become increasingly negative,

indicating an increasingly curved left turn until a sharp return to the final transition similar to that shown in Figure 7 as the curvature function discontinuously returns to 0. The wheel speeds also show a similarity to the first transition of Figure 7 but instead of a steady state following the initial transition, the speeds of all wheels start to decrease as the curvature of the path increases (hence radius of curvature decreases) followed by a sharp return to a final transition similar to that shown in Figure 7 as the curvature function discontinuously returns to 0.

3.4 Simulation with two segments and variable speed and curvature

Up to this point the simulations have used simple combinations of prescribed curvature and velocity for the first hitch point so that the resulting plots can be easily interpreted as evidence the algorithm is working correctly. In this section we simulate the more complex system of a sinusoidally varying curvature (see equation (10)), a velocity that has a parabolic section decreasing from 1 m/s to 0.5 m/s and then increasing to 1 m/s again (see equation (11)) and two segments of the snake-like robot. Ten simulation runs using $\Delta t = 0.001$ gave an average CPU time of $0.00047 = 0.47\Delta t$ (on a laptop with *Windows* XP operating system, 2.96 GB RAM, processor speed 2.53 GHz) so that the sampling time far exceeds the calculation time.

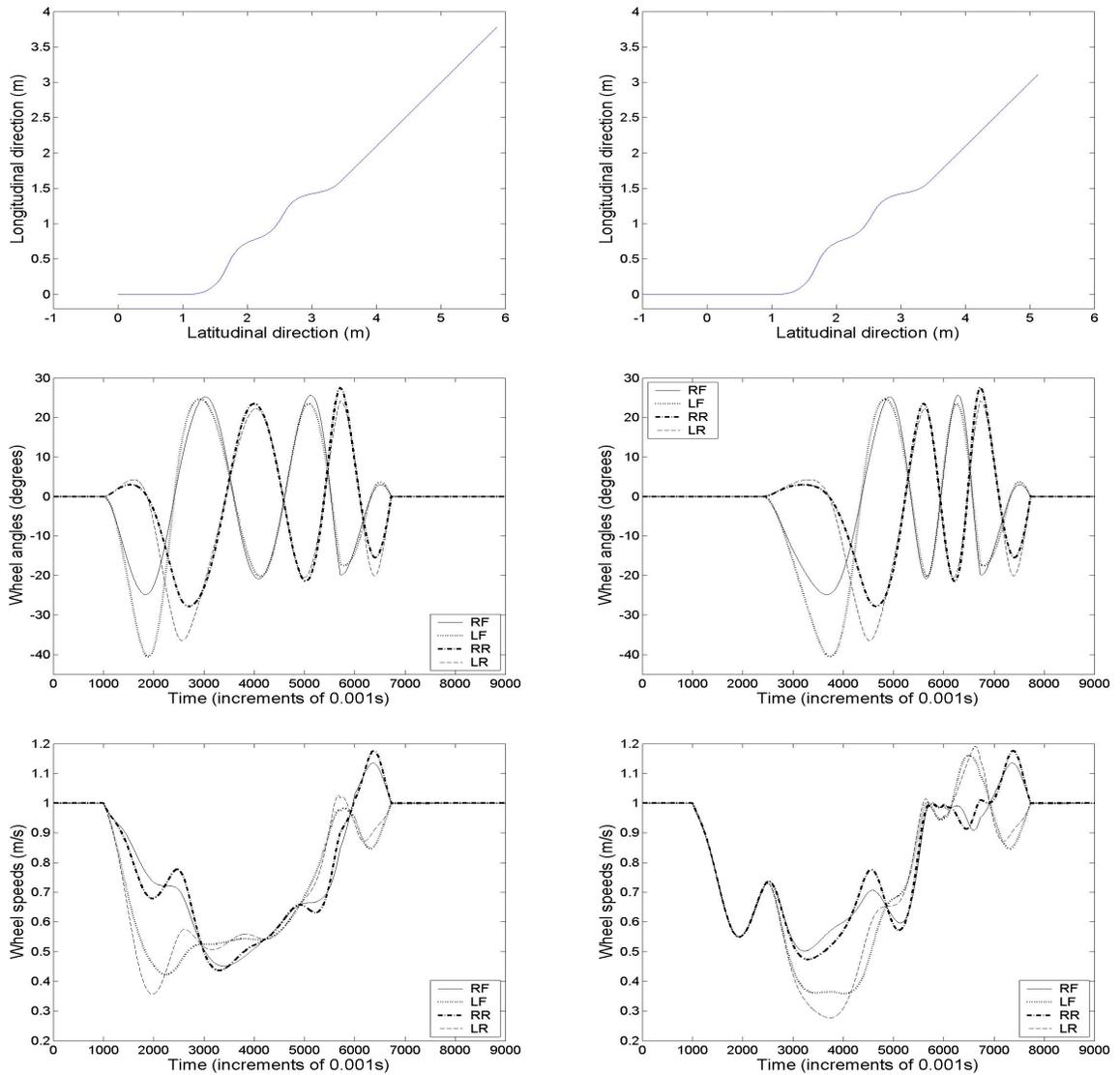


Figure 11: Two segment simulation with varying speed and curvature. The left column shows the first segment's front hitch path, wheel angles and wheel speeds. The right column shows the second segment's front hitch path (rear hitch path of first segment), wheel angles and wheel speeds.

$$\kappa(t) = \begin{cases} 0 & 0 < t \leq 1 \\ 2 \sin(3(t-1)) & 1 \leq t \leq \frac{3}{2}\pi + 1 \\ 0 & \frac{3}{2}\pi + 1 \leq t \leq 9 \end{cases} \quad (10)$$

$$v(t) = \begin{cases} 1 & 0 < t \leq 1 \\ \frac{2}{25}x^2 - \frac{14}{25}x + \frac{37}{25} & 1 \leq t \leq 6 \\ 1 & 6 \leq t \leq 9 \end{cases} \quad (11)$$

The hitch path of the first and second hitch as well as the wheel angles and speeds of the two modules are plotted in Figure 11. With such a complex set of conditions interpretation of the results is more difficult. Reassurance that the algorithm is producing the correct outputs is offered by the following set of observations. Firstly the first and second hitch points follow the same path with the second starting and finishing 1 m in a straight line behind the first. This is because $\kappa(t)$ was chosen to start and finish with straight sections of zero curvature. There is an expected lag between the wheel angles of front and rear wheels with the second segment showing a near identical pattern with a slight time dilation in places (such as the section between 3 and 5 second) when the velocity is slowing. There is a noticeable overall slowing of wheels as the front hitch velocity is decreased which then speed back up as the $v(t)$ returns to a value of 1 m/s. As expected although $v(t)$ returns to 1 m/s at $t = 6$ s, the wheels do not all return to a speed of 1 m/s until their angles are all

zero and the segment is again traveling in a straight line.

4 Conclusion

A new algorithm has been developed to accurately control the wheel angles and speeds of 4WD4WS robots with the added prescription that reference points at front and rear traverse the same path. The accuracy has been tested by simulation compared to an analytical solution and also by achieving the expected results of further simulations with simple modifications of the control parameters. If the reference points at front and rear are hitching points of a segmented vehicle then the control system is applicable to a snake-like robot. Furthermore, if some segments are physically removed, because the hitch points are passive, the algorithm provides a control system for a series of path following robots (compare to [16]).

Modifications of this control algorithm, such as position of the reference points traversing a single path, will also be applicable to truck-trailer systems with autonomous steered and driven wheels (compare to [17]). The symmetry of a snake-like robot driven forward from the front hitch or in reverse from the rear hitch means the algorithm is also applicable to reversing truck-trailer systems with autonomous steered and driven wheels (compare to [18, 19]).

References

- [1] I. J. Spark and M. Yousef Ibrahim. Integrated mechatronics solution to maximize tractability and efficiency of wheeled vehicles. In Bruno Siciliano, editor, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 320 – 325, Como, Italy, July, 2001.
- [2] I. J. Spark and M. Yousef Ibrahim. Manoeuvrable gantry tractor comprising a chorus line of synchronised modules. *IEEE International Symposium on Industrial Electronics*, pages 2208 – 2213, 2007.
- [3] S. Hirose and H. Yamada. Snake-like robots: Machine design of biologically inspired robots. *IEEE Robotics & Automation Magazine*, 16(1):88 – 98, 2009.
- [4] A. Percy, I. J. Spark, and M. Yousef Ibrahim. On-line determination of wheel angles and speeds for manouevrable gantry tractor comprising a “chorus line” of synchronised maodules. In *IEEE-ICIT'09 International Conference on Industrial Technology*, pages 320 – 325, Churchill, Australia, February, 2009.
- [5] E. Slawiński, V. Mut, and J. F. Postigo. Teleoperation of mobile robots with time-varying delay. *Robotica*, 24:673 – 681, 2006.

- [6] B. Klassen and K. Paap. GMD-SNAKE2: A snake-like robot driven by wheels and a method for motion control. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 3014 – 3019, Detroit, Michigan, USA, 1999.
- [7] G. Campion, G. Bastin, and B. D’Andrea-Novel. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on robotics and automation*, 12(1):47 – 61, 1996.
- [8] Q. Han and L. Dai. A non-linear dynamic approach to the motion of four-wheel-steering vehicles under various operation conditions. In *Proceedings of the Institute of Mechanical Engineers, Part D: Journal of automobile engineering*, volume 222, pages 535 – 549, 2008.
- [9] H. Itoh, A. Oida, and M. Yamazaki. Numerical solution of a 4WD-4WS tractor turning in a rice field. *Journal of terramechanics*, 36:91 – 115, 1999.
- [10] Shou-Tao Peng. On one approach to constraining the combined wheel slip in the autonomous control of a 4WS4WD vehicle. *IEEE Transactions on control systems technology*, 15(1):168 – 175, 2006.
- [11] I. Waheed and R. Fotouhi. Trajectory and temporal planning of a wheeled mobile robot on an uneven surface. *Robotica*, 27:481 – 489, 2009.

- [12] G. Klančar, D. Matko, and S. Blažič. Wheeled mobile robots in a linear platoon. *Journal of Intelligent robotic systems*, 54(5):709 – 731, 2008.
- [13] K. Watanabe, J. Yamakawa, M. Tanaka, and T. Sasaki. Turning characteristics of multi-axle vehicles. *Journal of Terramechanics*, 44(1):81–87, 2007.
- [14] G. Scaglia, L. Q. Montoya, V. Mut, and F. di Sciascio. Numerical methods based controller design for mobile robots. *Robotica*, 27:269 – 279, 2009.
- [15] M Lipschutz. *Theory and problems of differential geometry*. McGraw-Hill, 1969.
- [16] M. Chueh, Y. L. W. A. Yeung, Kim-Pang C. Lei, and S. S. Joshi. Following controller for autonomous mobile robots using behavioral cues. *IEEE Transactions on Industrial Electronics*, 55(8):3124 – 3132, August 2008.
- [17] K. Rangavajhula and H. S. J. Tsao. Active trailer steering control of an articulated system with a tractor and three full trailers for tractor-track following. *International Journal of Heavy Vehicle Systems*, 14(3):271–293, 2007.
- [18] K. Matsushita and T. Murakami. Nonholonomic equivalent disturbance based backward motion control of tractor-trailer with virtual steering. *IEEE Transactions on Industrial Electronics*, 55(1):280 – 287, January 2008.

- [19] K. Tanaka, K. Yamauchi, H. Ohtake, and Hua O. Wang. Sensor reduction for backing-up control of a vehicle with triple trailers. *IEEE Transactions on Industrial Electronics*, 56(2):497 – 509, February 2009.