IEEE Transactions on Robotics and Automation, February 1995, Vol. 11, No. 1, pp. 21-35.

CONTROL AND KINEMATIC DESIGN OF MULTI-DEGREE-OF-FREEDOM MOBILE ROBOTS WITH COMPLIANT LINKAGE

by
Johann Borenstein
The University of Michigan
Department of Mechanical Engineering and Applied Mechanics
Ann Arbor, MI 48109-2110
Ph.: (313) 763-1560

ABSTRACT

Multi-degree-of-freedom (MDOF) vehicles have many potential advantages over conventional (i.e., 2-DOF) vehicles. For example, MDOF vehicles can travel sideways and they can negotiate tight turns more easily. In addition, some MDOF designs provide better payload capability, better traction, and improved static and dynamic stability. However, MDOF vehicles with more than three degrees-of-freedom are difficult to control because of their overconstrained nature. These difficulties translate into severe wheel slippage or jerky motion under certain driving conditions. These problems make it difficult to benefit from the special motion capabilities of MDOF vehicles in autonomous or semi-autonomous mobile robot applications. This is so because mobile robots (unlike *automated guided vehicles*) usually rely on dead-reckoning between periodic absolute position updates and their performance is diminished by excessive wheel slippage.

This paper introduces *compliant linkage*, a new concept in the control and kinematic design of MDOF mobile robots. As the name implies, *compliant linkage* provides compliance between the drive wheels or drive axles of a vehicle, to accommodate control errors which would otherwise cause wheel slippage.

The concept of *compliant linkage* was implemented and tested on a 4-DOF vehicle built at the University of Michigan's Mobile Robotics Lab. Experimental results show that control errors are effectively absorbed by the linkage, resulting in smooth and precise motion. The dead-reckoning accuracy of the *compliant linkage* vehicle is substantially better than that reported in the literature for other MDOF vehicles; it was found equal to, or even better than that of comparable 2-DOF vehicles.

1. INTRODUCTION

Most conventional mobile robots use either a *tricycle* design where one wheel is steered and driven [13, 22] or a *differential drive* design (i.e., two drive wheels, each with its own motor [18]). Differential drive vehicles have the added advantage that they can turn in place. Either kind of vehicles has two independently controlled axis and is easy to control. However, neither design allows sideways motion.

One smart design that does allow sideways motion is the so-called *synchro-drive* [23]. Synchro-drive vehicles typically have three driven and steered wheels that are mechanically linked to one drive motor and one steer motor (i.e., these vehicles are still 2-DOF). The three wheels can be steered into any direction, but are parallel to each other at all times. This design allows the vehicle to move in all directions, but there is no control over the orientation of the vehicle body (since only the wheels turn). While this problem can be overcome by adding an independently controlled turret platform, the synchro-drive design would still be suitable only to round (or approximately round) vehicles. Yet, many transport tasks require large cargo capacities that are best provided by rectangular-shaped mobile robots, which can traverse narrow aisles in factories.

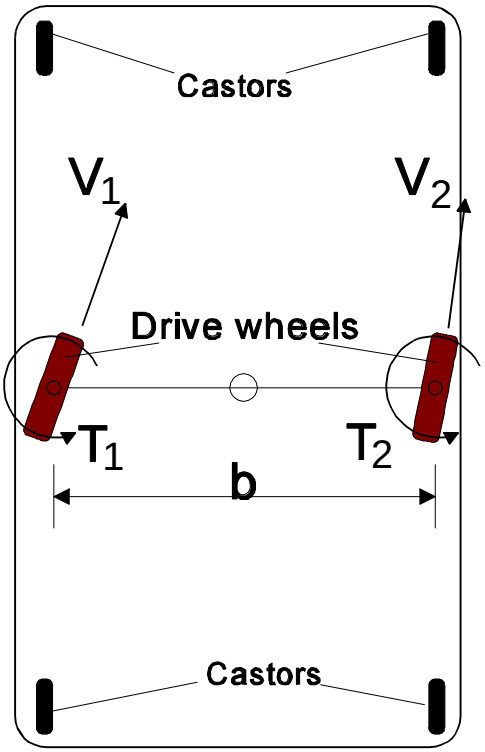
A design suitable for rectangular platforms employs special wheels that can roll sideways [16, 11, 14, 15]. Such vehicles, usually driven by three or four independent drive motors, are useful in some applications but cannot be used efficiently on any but smooth and regular surfaces [11]. Since most industrial applications don't provide such smooth surfaces, we will limit the following discussion to *multi-degree-of-freedom* (MDOF) vehicles with full-sized, “conventional” wheels.

MDOF vehicles are ideal for transport tasks in confined space. Theoretically, MDOF vehicles are extremely maneuverable; they are capable of turning in confined space, moving sideways, and performing other maneuvers that would allow the vehicle to move along a mathematically optimal trajectory. A good MDOF design could significantly reduce the amount of floor space required for safe vehicle operation.

Although a vehicle with more than two independently controlled axis offers exceptional advantages in terms of maneuverability, it also causes exceptional difficulties in terms of control. These difficulties are discussed in detail in Section 2. Section 3 introduces the concept of *compliant linkage* and presents two different 4-DOF designs that implement *compliant linkage*. Section 4 describes the control system and Section 5 presents experimental results.

2. BACKGROUND

One typical design of an MDOF vehicle is the *four-degree-of-freedom* (4-DOF) vehicle shown in Fig. 1. An actually existing prototype vehicle based on this design is HERMIES-III, a vehicle which was developed and built at the Oak Ridge National Laboratory (ORNL) as part of an ongoing research program of the Department of Energy. HERMIES-III has two tricycle drives, with a total of four motors. Four castors at the vehicle corners provide stability. Even though HERMIES-III is a very advanced and exceptionally well-designed system, researchers at ORNL [19, 20] reported on large position errors after certain maneuvers, caused by severe wheel slippage.



(1)**Figure 1:**
Four-degree-of-freedom vehicle.

Another account of difficulties with the control of a multi-degree-of-freedom vehicle was given by Hans Moravec, one of the leading researchers in Mobile Robots. In a technical report Moravec [17] described problems with PLUTO, a 6-DOF vehicle developed at Carnegie-Mellon University. In this report Moravec observed:

> *"... severe oscillations and other errors in servoing the drive and steering motors."*

and that

> *"With all [motor assemblies] running the robot mostly shook and made grinding noises."*

A thorough analysis of the nature of these problems revealed that they could be remedied by introducing a novel kinematic design and control system. Before we present such a design in Section 3 we will discuss some of the problems in more detail.

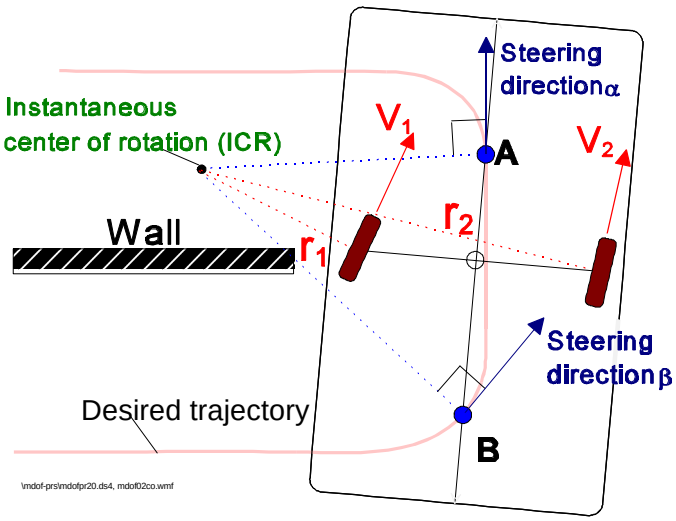
2.1 The instantaneous center of rotation (ICR) for trajectory control

One effective way to control the trajectory of a MDOF vehicle is based on the concept of the *instantaneous center of rotation* (ICR). Although this method is not new [1; 10; 20], it is described here to illustrate typical requirements for a MDOF vehicle.

For the purpose of discussing the ICR concept we will assume that points **A** and **B** on the vehicle should *momentarily* travel in the directions α and β, as indicated in Fig. 2. A trajectory like the one in Fig. 2 can be prescribed by a *guide-wire* in applications for *automated guided vehicles* (AGVs), or by an obstacle avoidance system in applications for *autonomous mobile robots (*AMRs) [3].

The ICR concept is borrowed from the areas of machine design and kinematics: it is an imaginary point around which a rigid body appears to be rotating *momentarily* (for an instant

dt), when the body is rotating and translating. In pure translatory motion, the ICR is located at a distance ∞ from the body. One special case of translatory motion exists when both wheels are parallel to the longitudinal axis of motion. This configuration corresponds to the widely used *differential drive*, where two wheels are located on the same axes but are driven by individual motors. We will call this the *normal configuration*, and, by contrast, we will use the term *crabbing* when at least one wheel is not oriented parallel to the longitudinal axes of the vehicle.



(1)**Figure 2:** Controlling a 4-DOF vehicle by *Instantaneous Center of Rotation*.

For the vehicle in Fig. 2, The ICR is constructed as the crosspoint of the two normals to the steering directions α and β. Then, the orientation of the two wheels is set normal to the two position vectors r_1 and r_2. Clearly, this orientation of the drive-wheels *will* cause rotation around the ICR and, consequently, rotation around the ICR results in points **A** and **B** *momentarily* moving in the required steering directions. However, the velocities of the wheels must maintain the ratio

$$\frac{V_1}{V_2} = \frac{r_1}{r2} \qquad (1)$$

Equation (1) is also known as the "*rigid body constraint*."

Note that V_1 will be *independent* from V_2 when $r_1 = r_2 = \infty$ (i.e., in *normal* configuration). It is also important to point out that the ICR concept can be applied to vehicles with any number of degrees of freedom (e.g., 4 drive/4 steer kinematics).

Any rigid-body vehicle with more than 3 degrees-of-freedom must maintain Eq. (1) accurately, that is, the ratio between the drive velocities must be maintained and match the momentary angles α and β, for otherwise *wheel slippage* will occur (see formal proof in [1]).

2.2 Wheel slippage in MDOF vehicles

In this Section we will discuss the physical significance of the *rigid body constraint*, and how different causes for violation of the constraint translate into wheel slippage.

In practice, one finds that Eq. (1) is particularly difficult to maintain, because the relative location of the ICR usually changes between sampling intervals, requiring instantaneous

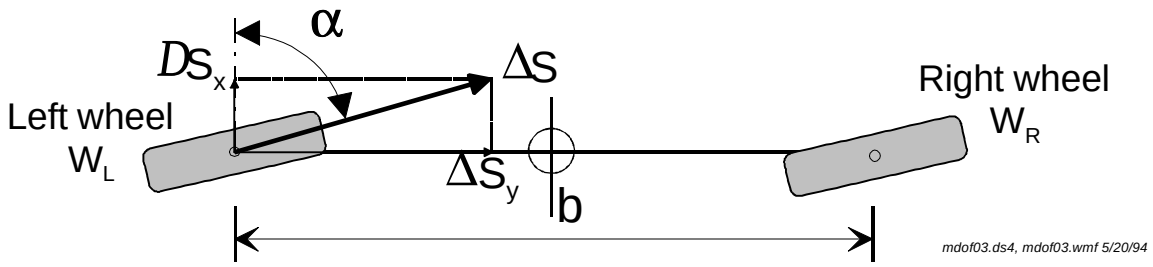
velocity changes for all drive and steering motors. Unfortunately, conventional DC-motor velocity control loops cannot *precisely* follow the prescribed velocity profile during transients. The integral of these errors over time translates into *permanent* (not transient) position errors for each wheel. In other words, even if the control loops for the drive motors had the same velocity *reference* commands, each loop would still end up with different pulse counts from the encoders.

For the simple case of the *normal* configuration, the pulse count difference Δs causes an orientation error $\Delta\theta$ that can be expressed as:

$$\Delta\theta = \frac{\Delta s}{b} \tag{2}$$

If not corrected immediately, $\Delta\theta$ may cause a very large *lateral* position error during subsequent motion. Even under steady-state conditions, mechanical disturbances on the wheels (e.g., bearing friction) will cause the wheels to rotate temporarily at different speeds and therefore generate different numbers of pulses. In conventional 2-DOF vehicles one may consider the resulting orientation error "permissible," since it does not cause slippage. In MDOF vehicles, on the other hand, the difference in pulse counts Δs will cause not only an orientation error $\Delta\theta$, but also wheel slippage. Wheel slippage causes much more severe problems than do position errors caused by inadequate control. This is so because wheel slippage errors cannot be corrected after they occurred, while control-type errors can.

Figure 3 visualizes the effect of Δs in a 4-DOF vehicle during *crabbing*. We assume that the right wheel, W_R, is stationary while the left wheel, W_L, rotates through the extra distance Δs. Decomposing Δs into its orthogonal components $\Delta s_x = \Delta s \sin\alpha$ and $\Delta s_y = \Delta s \cos\alpha$, it is clear from Fig. 3 that Δs_y is **completely lost to slippage** because the distance between the two drive wheels is physically fixed.



(3) **Figure 3:** Problems with *crabbing* motion.

Beside the inherent control problem of maintaining accurate velocity ratios between the drive wheels during *crabbing*, additional slippage is likely to be introduced because of the error due to unequal wheel-diameters, as explained next.

Most mobile robots use rubber tires to increase traction. However, it is difficult to manufacture rubber wheels with exactly the same diameter. In addition, unequally distributed loads will slightly compress one wheel more than the other, thus changing its effective diameter. When the motor controllers translate wheel revolutions into linear travel distance, their computation is based on the *nominal* wheel diameter. Even a very small diversion from the nominal wheel diameter renders this computation inaccurate.

With a typical commercial mobile robot (*LabMate* [25]), we measured the ratio between the left and right wheel diameters $D_L/D_R = 0.991$, for an unloaded vehicle. In the 2-DOF differential drive *LabMate* vehicle, this non-unity ratio causes some position error, but no slippage. However, in the 4-DOF vehicle of Fig. 1, the problem of unequal wheel diameters will necessarily cause a violation of Eq. (1), even if the controllers were capable of zero-error motor-velocity control at all times. Theoretically, this problem is further exacerbated because once a constant rate of slippage exists, any addition lateral force component acting on the slipping wheel will cause addition slippage.

For this reason we conclude that a mechanical means for implementing *compliance* must be designed into any MDOF vehicle. The mechanical compliance can accommodate temporary deviations from the required velocity ratios (i.e., Eq. 1), until the controllers catch up to correct the problem. Another way to look at the compliance approach is that a vehicle with mechanical compliance is not a rigid body and therefore does not have to comply with Eq. (1).

Existing MDOF vehicles like PLUTO or HERMIES-III do not have an *intentionally* designed mechanical compliance. Consequently, those vehicles may either "rattle" and "shake" as they try to accommodate position errors Δs through *unintentional* compliance such as backlash, or they may suffer from extensive slippage.

3. IMPLEMENTATION OF COMPLIANT LINKAGE

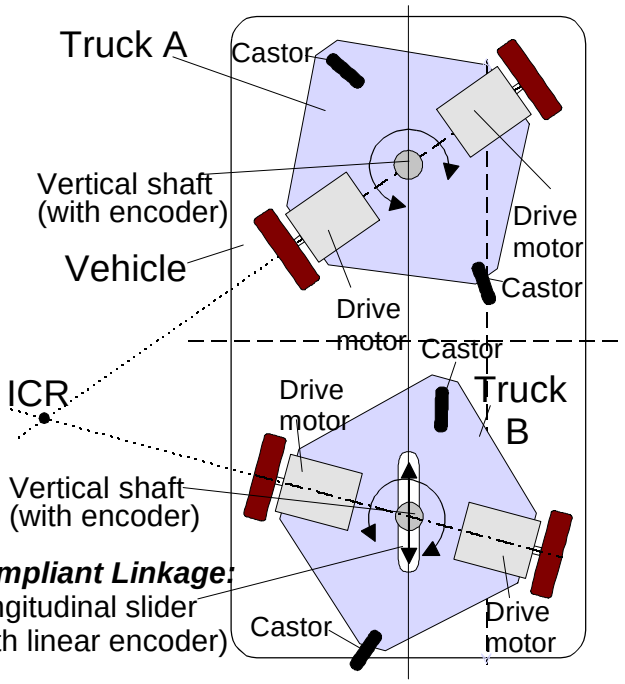
The key element in any workable MDOF design must be the provision of at least *some* mechanical compliance. In general, mechanical compliance may be realized by mounting all but one drive wheel such that they may slide freely in the desired direction of compliance. In practice, however, such an approach may be too expensive. Furthermore, a vehicle with more than four DOF does not add any motion capabilities beyond those that can be realized by a 4-DOF vehicle. For this reason we will limit our discussion to 4-DOF vehicles.

3.1 A four-DOF vehicles with compliant linkage

One possible kinematic design for an MDOF vehicle is shown in Fig. 4. This vehicle has two independent drive units or "*trucks*" that are free to rotate about a vertical shaft connected to the vehicle body. Each *truck* comprises two drive motors, along with their respective reduction gears, encoders, and drive wheels. Each pair of drive wheels is located on a common axes and forms

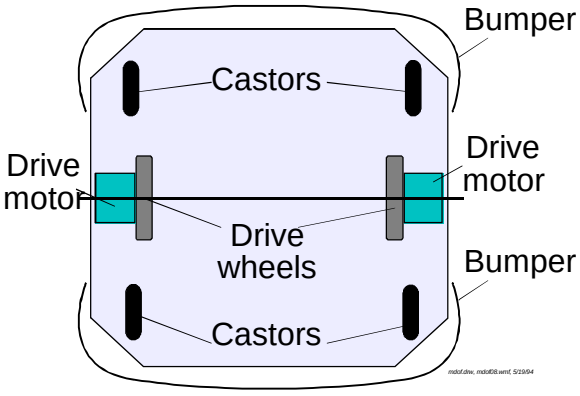
a *differential drive system* capable of moving forward, backward, and rotating) simply by controlling the velocities of the drive wheels. Each *truck* also holds two or more castors, for stability when traveling sideways. We will call a vehicle that combines two differential drive systems a *dual differential drive* (DDD) vehicle. Mechanical compliance is implemented by means of a linear bearing that allows relative motion between the front and rear truck. In the DDD vehicle, we will use the term *compliant linkage* for the compliant link that connects the two trucks.

Besides the encoders that are attached to each one of the drive motors, three additional encoders are needed: one rotary encoder on each of the two vertical shafts, and one linear encoder on the longitudinal slider.



(3)**Figure 4:** A *dual differential drive*, 4-DOF vehicle with *compliant linkage*.

One advantage of the DDD design in Fig. 4 over other 4-DOF configurations (see [4]) is its inherent **actuator redundancy**, that is, the ability to function in the event that **one motor** fails. In this case, the vehicle can be recovered by pulling the vehicle with the intact axle while steering the half-disabled axle with the remaining intact motor[1]. With this capability, the mobile robot can still perform many tasks, or, at the very least, retrieve **itself** from an operation. Actuator redundancy is particularly beneficial in *hazzardous environments*, such as in Nuclear Power Plants [9] and in Nuclear Waste Storage facilities [8].



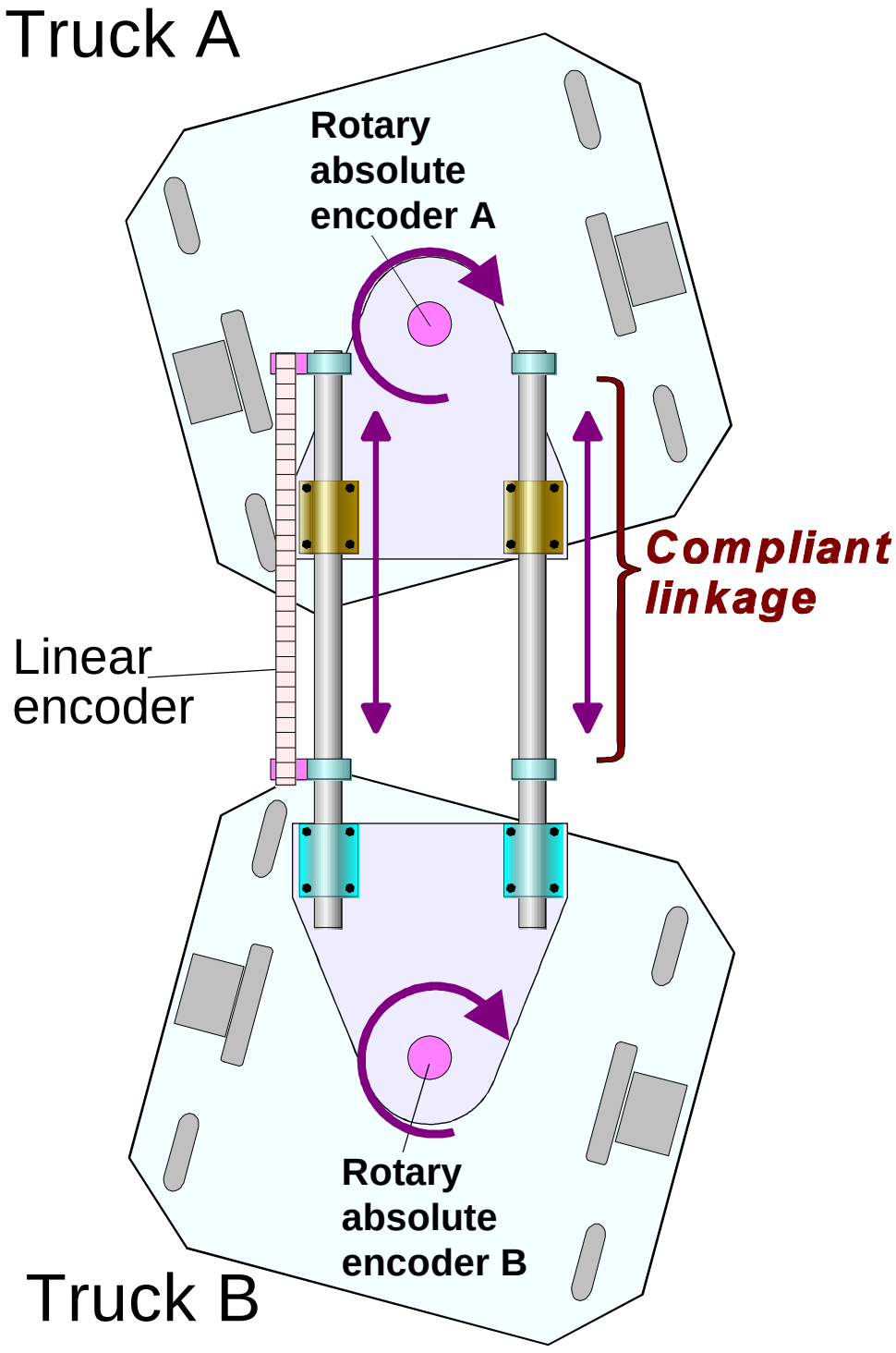
(3)**Figure 5:** Bottom view of the *differential drive LABMATE*® vehicle.

3.2 Implementation of the experimental vehicle

In order to verify the feasibility of the *compliant linkage* concept, a 4-DOF, DDD vehicle was built at the Mobile Robotics Lab at the University of Michigan. This experimental vehicle is constructed from two commercially available TRC *LABMATE*® platforms (see Fig. 5), connected by a sliding *compliant link* (see Fig. 6). Besides the total of four incremental encoders for the drive-

[1] We have implemented and experimentally verified this function on our experimental 4-DOF vehicle, which is described in Section 6.

wheels, the vehicle uses one incremental linear and two absolute rotary encoders, as shown i n Fig. 6.

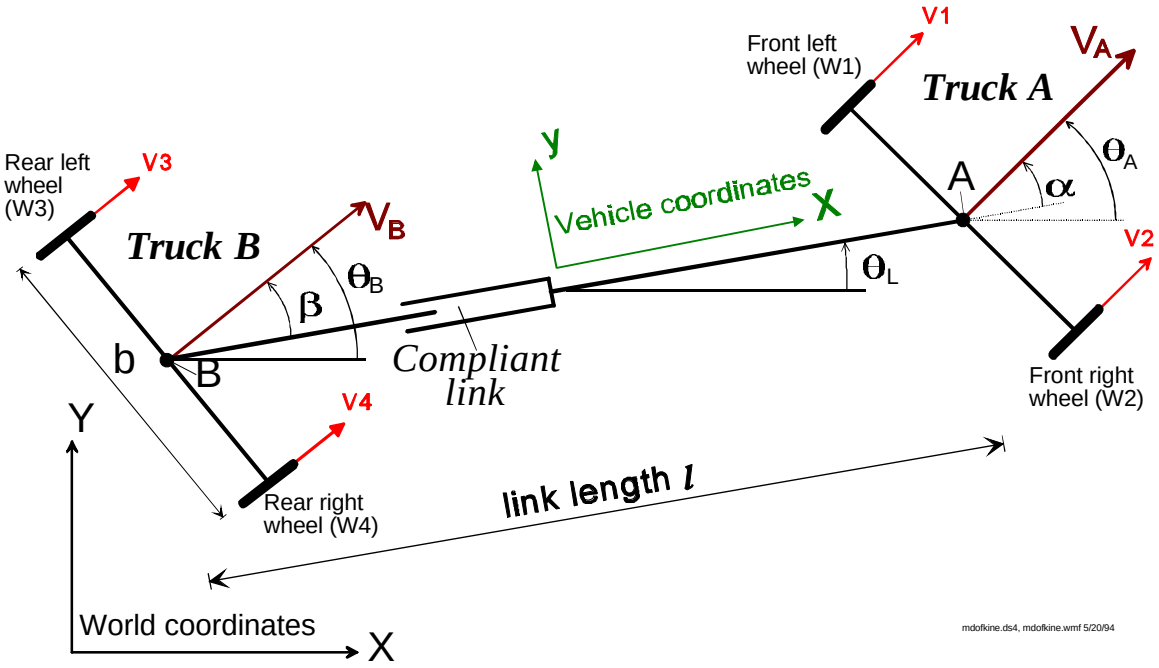


mdof09b.ds4, mdof09b.wmf

(3)**Figure 6:** 4-DOF prototype vehicle based on two *LABMATE*® platforms.

4. KINEMATIC ANALYSIS

This section analyzes the kinematics of the 4-DOF *dual differential drive* vehicle of Fig. 6. The purpose of the analysis is to define the relationship between all known or measurable positions and velocities, and all remaining relevant (but unknown) positions and velocities (see Fig. 7) . Note that the origin of the vehicle coordinate system in Fig. 7 is defined at the midpoint between truck centers A and B.



(3) **Figure 7:** Kinematic quantities for the *dual differential drive* 4-DOF vehicle.

Known parameters

N - Number of encoder pulses per wheel revolution
b - Wheel-base
r - Wheel radius
T - Sampling time

Measured quantities

l - Momentary length of the compliant link.
$\dot{l}$ - Rate of change of link-length l.
$\Delta N_{1,2}$ - Number of encoder pulses produced by wheels W_1 and W_2 between t_{n-1} and t_n
α, β - Relative angle between the compliant link and trucks A and B, respectively.
$\dot{\alpha}, \dot{\beta}$ - Relative angular velocity of trucks A and B, with respect to the compliant link.

Quantities computed by direct kinematics

x_1, y_1, x_2, y_2	-	Absolute position of front wheels W_1 and W_2, in world coordinates.
x_3, y_3, x_4, y_4	-	Absolute position of rear wheels W_3 and W_4, in world coordinates.
x_A, y_A, θ_A	-	Absolute position of center point *A*, absolute orientation of truck A.
x_B, y_B, θ_B	-	Absolute position of center point *B*, absolute orientation of truck B.
θ_L	-	Absolute orientation of the compliant link.
V_A, V_B	-	Absolute velocity of center points A and *B* on trucks A and B.
V_1, V_2	-	Absolute velocity of front wheels W_1 and W_2.
V_3, V_4	-	Absolute velocity of rear wheels W_3 and W_4.
$\Delta U_{1,2}$	-	Distance travelled by wheels $W_{1,2}$ between t_{n-1} and t_n.
$\Delta\theta_{A,n}$	-	Change in orientation of truck A between t_{n-1} and t_n.
$\dot{\theta}_L$	-	Absolute angular velocity of the compliant link.
$\dot{\theta}_A, \dot{\theta}_B$	-	Absolute angular velocity of trucks A and B.

The first step in this analysis is the derivation of the momentary position of the wheels W_1 and W_2 (truck A) from a known starting position and from the incremental motion of the wheels as measured by the encoders (odometry). Note that the information from the whee l encoders of truck B is redundant and is completely discarded[2]. For this reason the position and velocities of truck B are listed under "computed quantities."

a . Odometry for Truck A

In this Section we will use indices *n* and *n*-1 to distinguish between the state of variables at the present sampling instant (t_n) and the previous sampling instant (t_{n-1}), respectively.

The curvelinear distance traveled by each wheel can be computed from the measured encode r counts and the known parameters of the motor transmission.

$$\Delta U_{1,2} = \left(\frac{2\pi r}{N}\right)\Delta N_{1,2} \tag{3}$$

The change in orientation (for small changes) is

$$\Delta\theta_{A,n} = \left(\frac{1}{b}\right)(\Delta U_1 - \Delta U_2) \tag{4}$$

and the change in translation is

$$\Delta U_{A,n} = \tfrac{1}{2}\,(\Delta U_1 + \Delta U_2) \tag{5}$$

The new orientation and position of truck A (for small changes) at instant t_n is given by

[2] As this paper goes to the printer, we have just begun to evaluate a new method for using the redundant information from all wheel encoders to detect and correct dead-reckoning errors [6].

$$\theta_{A,n} = \theta_{A,n-1} + \Delta\theta_{A,n} \tag{6}$$

$$x_{A,n} = x_{A,n-1} + \Delta U_{A,n} \cos\left(\frac{\theta_{A,n-1} + \theta_{A,n}}{2}\right) \tag{7a}$$

$$y_{A,n} = y_{A,n-1} + \Delta U_{A,n} \sin\left(\frac{\theta_{A,n-1} + \theta_{A,n}}{2}\right) \tag{7b}$$

and the velocity of the front wheels is

$$V_{1,2} = \left(\frac{\Delta U_{1,3}}{T}\right) \tag{8}$$

b. Positions

For the sake of clarity we will omit the index *n* in this and all subsequent Sections (e.g., $\theta_{A,n}$ becomes θ_A).

Subtracting the relative orientation α (measured by the absolute encoder mounted on shaft A) from the above computed absolute orientation θ_A, the absolute orientation θ_L of the compliant link can be computed

$$\theta_L = \theta_A - \alpha \tag{9}$$

Similarly, the absolute orientation of truck B is found

$$\theta_B = \theta_L + \beta \tag{10}$$

The absolute position of center point B on truck B is

$$\begin{aligned} x_B &= x_A - l\cos\theta_L \\ y_B &= y_A - l\sin\theta_L \end{aligned} \tag{11}$$

Note that the position of the rear wheels W_3 and W_4, denoted (x_3, y_3) and (x_4, y_4), is of no importance and is omitted here.

c. Velocities

The magnitude of the absolute velocity V_A of center point A is

$$V_A = \frac{V_2 + V_1}{2} \tag{12}$$

and the rotational velocity is

$$\dot{\theta}_A = \frac{V_2 - V_1}{b} \tag{13}$$

The rotational velocity of the compliant link, $\dot{\theta}_L$, and of the rear truck, $\dot{\theta}_B$, are found by differentiation of Eqs. (9) and (10).

$$\dot{\theta}_L = \dot{\theta}_A - \dot{\alpha} \tag{14}$$

$$\dot{\theta}_B = \dot{\theta}_L + \dot{\beta} \tag{15}$$

Differentiation of Eqs. (11) yields the *x* and *y* velocity components of center point B (note that the length *l* of the compliant link is a variable with respect to time).

$$\begin{aligned} V_{Bx} &= V_A \cos\theta_A - \dot{l}\cos\theta_L + l\dot{\theta}_L \sin\theta_L \\ V_{By} &= V_A \sin\theta_A - \dot{l}\sin\theta_L - l\dot{\theta}_L \cos\theta_L \end{aligned} \tag{16}$$

The magnitude of the velocity of truck B is

$$V_B = \sqrt{V_{Bx}^2 + V_{By}^2} \tag{17}$$

Rewriting Eqs. (12) and (13) for the rear truck and solving simultaneously, the absolute wheel velocities for the rear wheels can be found.

$$\begin{aligned} V_3 &= V_B + \frac{b}{2}\dot{\theta}_B \\ V_4 &= V_B - \frac{b}{2}\dot{\theta}_B \end{aligned} \tag{18}$$

5. THE CONTROL SYSTEM

The control system comprises three levels, as shown in Fig. 8 [5]. The function of each level is discussed below.

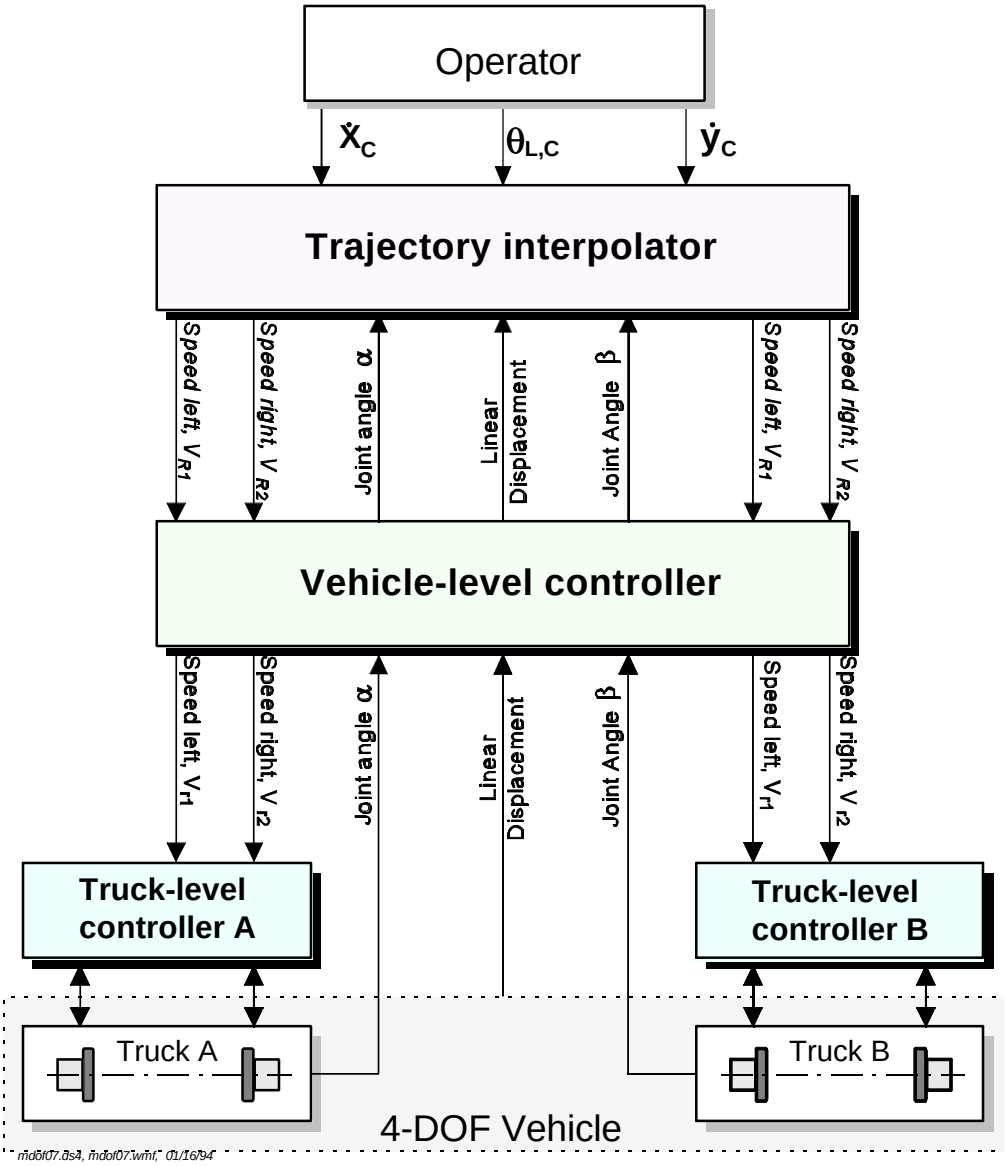


Figure 8: Major components of the MDOF vehicle control system.

5.1 The truck-level controller

At the lowest level of the controller hierarchy is the truck-level controller. The purpose of this controller is to maintain the velocities of each drive wheel, according to reference velocities prescribed by the vehicle-level controller. The truck level controller has an inner velocity feedback loop, which uses the commercially available, programmable *HCTL-1100* motion controller chip

[24]; one for each motor. These chips perform quadrature decoding of the incremental wheel encoder output, compute the actual velocity of the motors, V_m, and compare this velocity with the reference velocity V_r prescribed by the outer control loop. The difference $E = V_r - V_m$ is the error signal. Programmed as P-type controllers, the chips then issue *pulse width modulation* (PWM) signals to the PWM amplifiers, in proportion to the computed error E. The inner loop performs at a sampling time of T_i = 4 ms. The outer loop of the truck-level controller is a modified implementation of the cross-coupled controller developed earlier by Borenstein and Koren [2] for accurate control of differential drive mobile robots. The purpose of cross-coupling is to maintain an accurate ratio between the velocities of the two drive motors in a differential drive vehicle. The overall effect of the cross-coupled control is the elimination of steady-state orientation error of a truck, while allowing steady-state errors in the translational velocity of the truck center. This error is of less concern, since it is detected and corrected by the vehicle-level controller.

5.2 The vehicle-level controller

The vehicle-level controller is the central element in our system; its task is to minimize deviations Δl from the nominal link-length L (i.e., the length of the *compliant link* that connects the two trucks). The link-length is a function of the speed of each truck ***and*** its orientation relative to the link. This dual dependency creates a difficulty that can be visualized by considering the following two extreme cases:

Case a: both trucks are aligned longitudinally (i.e., $|\alpha| = |\beta| = 0^o$)
In this case, the link-length can only be controlled by changing the translational speed of the trucks.

Case b: both trucks are facing 90^o sideways (i.e., $|\alpha| = |\beta| = 90^o$)
In this case, the relative speed between the two trucks is always zero, and the link-length can only be controlled if we first change the orientation of the trucks.

In actual operation one will encounter a combination of these two extreme cases. The resulting control problem is rather difficult; it requires that we control the link-length by manipulating four motor velocities in a system where two basically different control laws apply (depending on the angles α and β) and where one of the control laws is highly non-linear (Case b, above).

To solve this problem, we first defined a simple *PI controller module*, to guarantee zero deviation from the nominal link-length under steady state conditions:

$$M_{PI} = K_P \Delta l + K_I \Sigma \Delta l \qquad (19)$$

where
M_{PI} — Output of the PI controller.
K_P — Proportional gain of the PI controller.
K_I — Integral gain of the PI controller.

Δl — Deviation from the nominal link-length L.

For Case (a) situations the *PI controller module* could be used as follows

$$V_{r,1} = V_{R,1}(1 - M_{PI}) \quad (20a)$$
$$V_{r,2} = V_{R,2}(1 - M_{PI}) \quad (20b)$$
$$V_{r,3} = V_{R,3}(1 + M_{PI}) \quad (20c)$$
$$V_{r,4} = V_{R,4}(1 + M_{PI}) \quad (20d)$$

where

$V_{R,i}$ — Required reference speed as received from the interpolator level.
$V_{r,i}$ — Modified reference speed, passed-on to the truck-level controller.

The link-length controller described by Eqs. (20) works as follows: Suppose at some instant the leading truck A is faster than the trailing truck B. This situation would result in an increased link-length l, or a positive Δl, and consequently a positive M_{PI}. Equations (20) then modify the reference velocities such that the speed of the leading truck is reduced, while the speed of the trailing truck is increased.

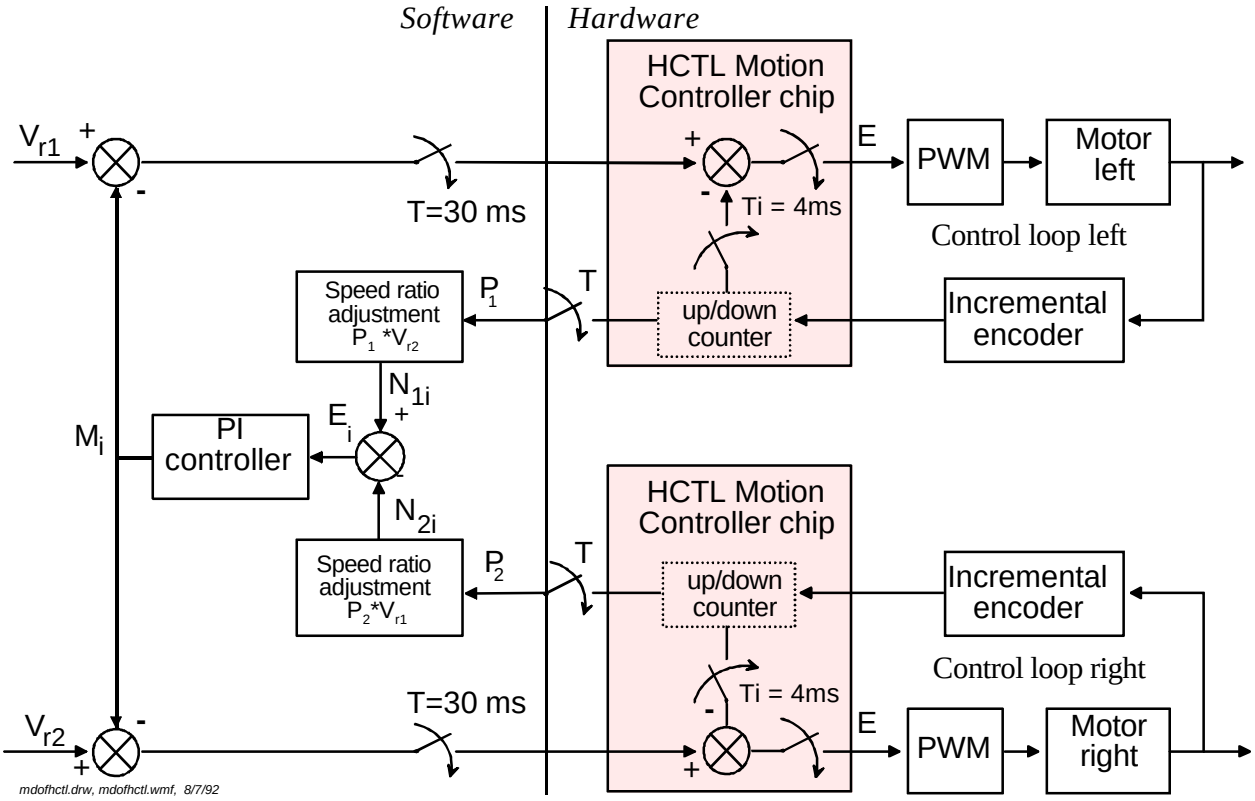
This simple link-length controller works well only if the two trucks are aligned longitudinally (i.e., $|\alpha|$ and $|\beta|$ are small). If $|\alpha|$ or $|\beta|$ are large, then modifying the speed of a truck is less effective. During fully 90° sideways crabbing, this control is in fact totally ineffective. If no additional measures are taken, then a small deviation from the link-length can grow over time and cause the controller to induce large (but ineffective) changes in the velocities of the trucks, resulting in instable motion.

For this reason we introduce a modification to the link-length controller. The modified controller applies a rotational correction (by increasing the difference between the velocities of the motors of a truck) when $|\alpha|$ or $|\beta|$ are large, while reducing the gain of the translational (i.e., the sum of the velocities of the motors on one truck) component. Equations (21) show how the modified controller is implemented.

$$V_{r,1} = V_{R,1}\ [1 + M_{PI}\ (-\cos\alpha\ -\ \sin\alpha)] \quad (21a)$$
$$V_{r,2} = V_{R,2}\ [1 + M_{PI}\ (-\cos\alpha\ +\ \sin\alpha)] \quad (21b)$$
$$V_{r,3} = V_{R,3}\ [1 + M_{PI}\ (+\cos\beta\ +\ \sin\beta)] \quad (21c)$$
$$V_{r,4} = V_{R,4}\ [1 + M_{PI}\ (+\cos\beta\ -\ \sin\beta)] \quad (21d)$$

As an example, we consider a typical situation like the one shown in Fig. 10a, where $\alpha = 75°$, $\beta = 20°$, and $M_{PI} > 0$ (i.e., the compliant link is extended beyond its nominal length). Applying the modified controller of Eqs. (21), we note that $\cos\alpha$ is small, $\sin\alpha$ is large, and the sum of the two trigonometric terms in Eq. (21a) is negative. Consequently, $V_{r,1}$ will be reduced. Similarly, $V_{r,2}$ will be increased, since the dominant $\sin\alpha$ term has a positive sign. Thus, the absolute speed *difference* between the two drive wheels of the front truck increases, modifying the motion prescribed by the *trajectory interpolator* (the highest level in the control hierarchy, explained in Section 5.3). This results in the addition of a counter-clockwise rotational component, while

slightly reducing the translational velocity of the truck (see Fig. 10b). The counter-clockwise rotation orients the leading truck more toward the trailing truck, thereby reducing the relative speed between the two trucks and consequently the link-length. The effect of the controller on the trailing truck can be examined in a similar way. In our example $\beta = 20^o$ is small, and therefore $\cos\beta$ is large. Since this term appears with a positive sign in Eqs. (21c) and (21d), it causes an increase in the translational velocity of the trailing truck, as shown in Fig. 10b. Consequently, the trailing truck can "catch up" with the leading truck and the link length is reduced. Note that the $\sin\beta$ terms in Eqs. (21c) and (21d) have opposite signs, causing a clockwise rotation. A clockwise rotation helps align the trailing truck with the compliant link, making the speed increase more effective. However, the rotational component in this example is only small, since $\sin\beta$ is small.



(19) **Figure 9:** Chassis-level controller with modified cross-coupling and HCTL motion controller.

It should be noted that the angles α and β were defined in Fig. 7 such that Eqs. (21) work out for all four quadrants. Furthermore, we only need to change the signs of the sine-terms to allow the same control for a truck moving backward. The trajectory interpolator (see Section 5.3) makes extensive use of this feature. For example, when the vehicle crabs to the right and is then instructed to crab to the left, it will simply reverse the motor voltage of either one (or both trucks, as needed), instead of turning the truck around 180^o. Equations (22) show the implementation of the controller for backward travel.

$$V_{r,1} = V_{R,1} \quad [1 + M_{PI} \quad (-\cos\alpha \quad + \quad \sin\alpha)] \qquad (22a)$$

$$V_{r,2} = V_{R,2} \quad [1 + M_{PI} \quad (-\cos\alpha \quad - \quad \sin\alpha)] \tag{22b}$$
$$V_{r,3} = V_{R,3} \quad [1 + M_{PI} \quad (+\cos\beta \quad - \quad \sin\beta)] \tag{22c}$$
$$V_{r,4} = V_{R,4} \quad [1 + M_{PI} \quad (+\cos\beta \quad + \quad \sin\beta)] \tag{22d}$$

5.3 The trajectory interpolator

The task of the *trajectory interpolator* is to generate reference velocity signals that direct the vehicle along a specific trajectory (for example, the trajectory shown in Fig. 2). There are many ways to implement a trajectory interpolator, and the ICR method described in Section 2.1 is only one of them. Another trajectory interpolator is described in this Section; it is designed for *tele-operator* control of the vehicle, but can be easily modified to allow computer-generated input.

5.3.1 Functional Description

The TI described here allows a human operator to control the vehicle motion with a 3-DOF joystick. The TI translates joystick control inputs $\dot{x}_C$ and $\dot{y}_C$ into linear Cartesian coordinate motion in vehicle coordinates (e.g., $\dot{x}_C$ causes pure forward travel and $\dot{y}_C$ causes pure sideways crabbing). The third control input, $\theta_{L,C}$, prescribes orientation. Our TI features an *alignment* option, where the $\theta_{L,C}$ control input is used to specify an *absolute* orientation with which the vehicle attempts to align at all times. This option is convenient for the operator when, for example, the vehicle travels through a narrow corridor, or when the vehicle emerges from a corridor with a known orientation γ, say, $\gamma = 90^o$, and then traverses an open workspace to dock with a station at $\gamma = 120^o$. The operator only needs to set the $\theta_{L,C} = 120^o$; the interpolator takes care of the alignment while the operator steers the vehicle toward the docking station, using only control inputs $\dot{x}_C$ and $\dot{y}_C$.

5.3.2 Implementation

The first operation of the TI after sampling all three joystick control inputs is to normalize diagonal joystick positions, so that any combination of $\dot{x}_C$ and $\dot{y}_C$ control inputs yields a maximum normalized value of 100.

$$\dot{x}_C{}' = \dot{x}_C \frac{*\dot{x}_C*}{\sqrt{\dot{x}_C{}^2 + \dot{y}_C{}^2}}$$
$$\dot{y}_C{}' = \dot{y}_C \frac{*\dot{y}_C*}{\sqrt{\dot{x}_C{}^2 + \dot{y}_C{}^2}} \tag{23}$$

where $\dot{x}_C{}'$ and $\dot{y}_C{}'$ are the control inputs, now normalized to a range of -100 to +100.

At this point it is convenient to define a new operator, which is called *angdist* and denoted (-). (-) is defined for two operands, α and β, and is used in the form $\Delta = \alpha$ (-) β. The result, Δ, is the shortest rotational distance between α and β. Therefore, *D* is always in the range $-180^o < \Delta < 180^o$.

Then, the TI determines from the $\theta_{L,C}$ control input the reference angular velocity $\dot{\theta}_{L,ref}$ *for the whole vehicle*.

$$\dot{\theta}_{L,ref} = K_{\theta} \frac{\theta_L \; (-) \; \theta_{L,C}}{T} \tag{24}$$

where
K_{θ} — Proportional gain factor for vehicle alignment.
θ_L — Present vehicle orientation.
T — Real-time sampling time.

Next, the velocity *x* and *y* components of the desired motion of the two trucks' center points A and B are determined.

$$\begin{aligned} V_{Ay} &= K_y \dot{y}_C{}' + \frac{1}{2} l \dot{\theta}_{L,ref} \\ V_{By} &= K_y \dot{y}_C{}' - \frac{1}{2} l \dot{\theta}_{L,ref} \end{aligned} \tag{25}$$

where
V_{Ay} — Velocity *y*-component of the front truck's center point, in *vehicle coordinate system*.
V_{By} — Velocity *y*-component of rear truck's center point, in *vehicle coordinate system*.
K_y — Proportional gain factor for dimensionless control input $\dot{y}_C$.

The normalized control input $\dot{x}_C{}'$ can be translated directly into the desired velocity components

$$\begin{aligned} V_{Ax} &= K_x \dot{x}_C{}' \\ V_{Bx} &= K_x \dot{x}_C{}' \end{aligned} \tag{26}$$

where
V_{Ax} — Velocity x-component of front truck center point, in vehicle coordinate system.
V_{Bx} — Velocity x-component of rear truck center point, in vehicle coordinate system.
K_x — Proportional gain factor for dimensionless control input $\dot{x}_C$.

Now, the TI can compute the magnitude of the reference velocities for the front and rear truck, $V_{A,ref}$ and $V_{B,ref}$

$$V_{A,ref} = \sqrt{V_{Ax}^2 + V_{Ay}^2}$$
$$V_{B,ref} = \sqrt{V_{Bx}^2 + V_{By}^2} \tag{27}$$

The directions of the reference velocities for the front and rear truck are

$$\alpha_{ref} = \arctan \frac{V_{Ax}}{V_{Ay}}$$
$$\beta_{ref} = \arctan \frac{V_{Bx}}{V_{By}} \tag{28}$$

To reach the desired reference directions α_{ref} and β_{ref} the front and rear trucks have to rotate. This is accomplished by applying the following reference steering rates:

$$\dot{\alpha}_{ref} = K_c \frac{\alpha - \alpha_{ref}}{T}$$
$$\dot{\beta}_{ref} = K_c \frac{\beta - \beta_{ref}}{T} \tag{29}$$

where
K_c — Proportional gain factor for steering.
α — Relative angle between front truck and link (measured by absolute encoder A).
β — Relative angle between rear truck and link (measured by absolute encoder B).
$\dot{\alpha}_{ref}$ — Reference steering rate for the front truck.
$\dot{\beta}_{ref}$ — Reference steering rate for the rear truck.

Rewriting Eqs. (14) and (15) we obtain

$$\dot{\theta}_{A,ref} = \dot{\theta}_L + \dot{\alpha}_{ref}$$
$$\dot{\theta}_{B,ref} = \dot{\theta}_L + \dot{\beta}_{ref} \tag{30}$$

Finally, rewriting Eqs. (18) for truck A and substituting Eqs. (30) the reference velocities for all four drive wheels are obtained.

$$
\begin{aligned}
V_{1,ref} &= V_A - \frac{b}{2}\dot{\theta}_{A,ref} \\
V_{2,ref} &= V_A + \frac{b}{2}\dot{\theta}_{A,ref} \\
V_{3,ref} &= V_B - \frac{b}{2}\dot{\theta}_{B,ref} \\
V_{4,ref} &= V_B + \frac{b}{2}\dot{\theta}_{B,ref}
\end{aligned}
\qquad (31)
$$

The velocities on the left-hand side of Eqs. (31) are the reference signals that are sent to the *vehicle-level controller*.

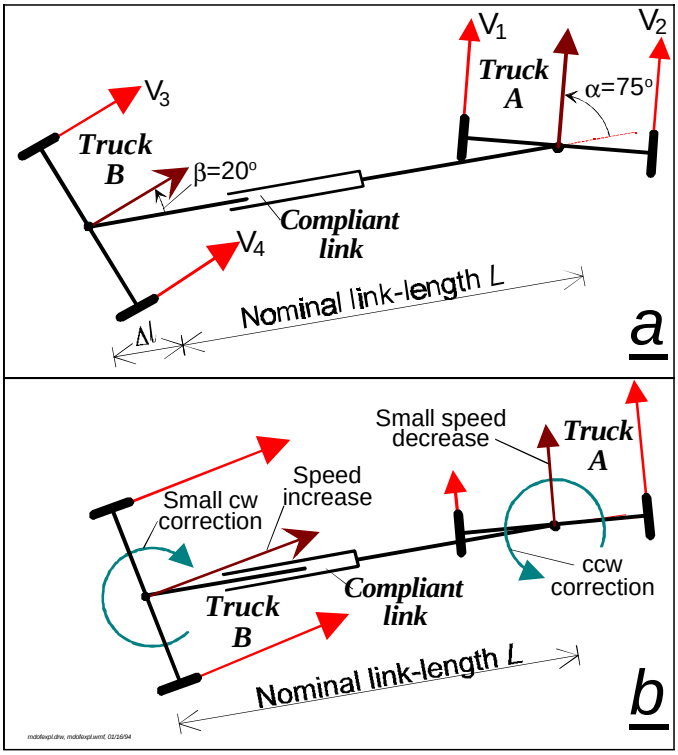
The trajectory interpolator performs several additional housekeeping functions, which are not explained here in detail. Among these functions are:

a. Imposing maximum limits on $\dot{\theta}_{L,ref}$.

b. Computing the difference between the current truck-orientation and the desired truck-orientation. If the directional difference is greater than 120°, a *reverse* flag is set and the motor velocities of the reversing truck are recomputed for backward travel. This function allows instantaneous and smooth transitions between forward and backward motion of each truck, as needed.

c. Reducing the translational speed of both trucks as a function of the reference steering $\dot{\alpha}_{ref}$ and $\dot{\beta}_{ref}$.

It should be noted that the trajectory interpolator does not provide a precise trajectory at higher speeds. For example, suppose the vehicle is in a position as shown in Fig. 10b and the operator suddenly prescribes a straight-forward direction at *full speed*. The trajectory interpolator will correctly align both truck *while* moving ahead, thereby causing an initially curved

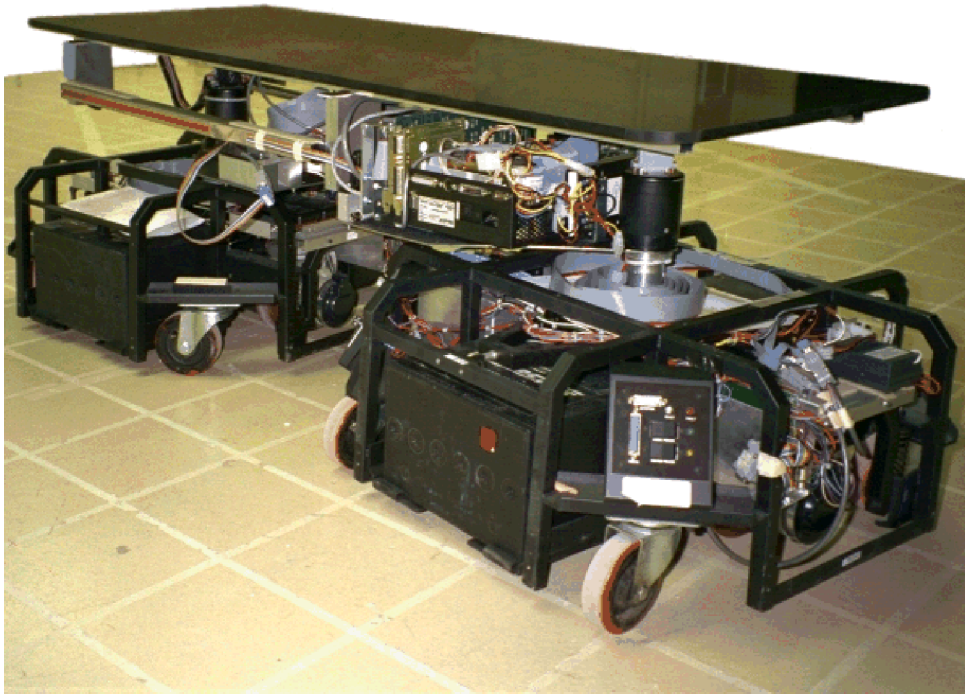


(32)**Figure 10:**
Example for the operation of the vehicle-level controller. a. link-length is too large. b. After corrective action.

path. If more precise trajectories are needed, then the operator (or a higher-level control algorithm) must keep the absolute reference velocities low during large changes of direction.

6. EXPERIMENTAL RESULTS

The dual differential drive vehicle was built and tested at the University of Michigan's Mobile Robotics Lab. Figure 11 shows the experimental vehicle, which is about 1.4 m long and 0.6 m wide. As described in Fig. 6, the vehicle is built from two *Labmate*® platforms, connected by the compliant linkage. On top of the vehicle, covering both platforms, is a plexiglass plate that provides continuous, flat loading space. A 486/33 MHz IBM-PC compatible computer onboard the vehicle implements all three levels of control at a sampling time of 30 ms (except for the hardware-controlled velocity feedback loop, which runs at 4 ms). The system is fully self-contained and draws power from four onboard batteries. A very convenient way of manually controlling the vehicle is by means of a commercially available 4-Channel FM radio control system, the kind that is normally used for model airplanes.



(32)**Figure 11:** The experimental dual differential drive vehicle, built and tested at the University of Michigan

The fundamental problem of existing MDOF vehicles is that the *actual* wheel velocities differ from the *required* wheel velocities, which are computed according to the kinematic constraints of the system (i.e., Eq. 1). Each violation of the kinematic constraints causes wheel slippage and, consequently, position errors. Reister [19] reports on orientation errors on the order of 20%, after certain maneuvers. It should be noted that in subsequent work Reister and Unseren [21]

developed a *Force Control Method* for their two-wheel drive/two-wheel steer platform, which resulted in a reported 20-fold improvement of accuracy. However, the experiments on which these results were based avoided *simultaneous* steering and driving of the two steerable drive wheels. This way, the critical problem of coordinating the control of all four motors *simultaneously and during transients* was completely avoided.

The basic claim of this paper is that the *compliant linkage* design eliminates such errors and the following experimental results support this claim. Of course, there are still other causes for position errors, which affect 2-DOF vehicles as well as MDOF vehicles. The most prominent of these "conventional" errors are:

- Unequal wheel diameters
- Misalignment of wheels
- Non-point contact area between wheels and floor
- Uneven floors
- Slippage during turning maneuvers

A more detailed discussion on some of these causes is given in [2].

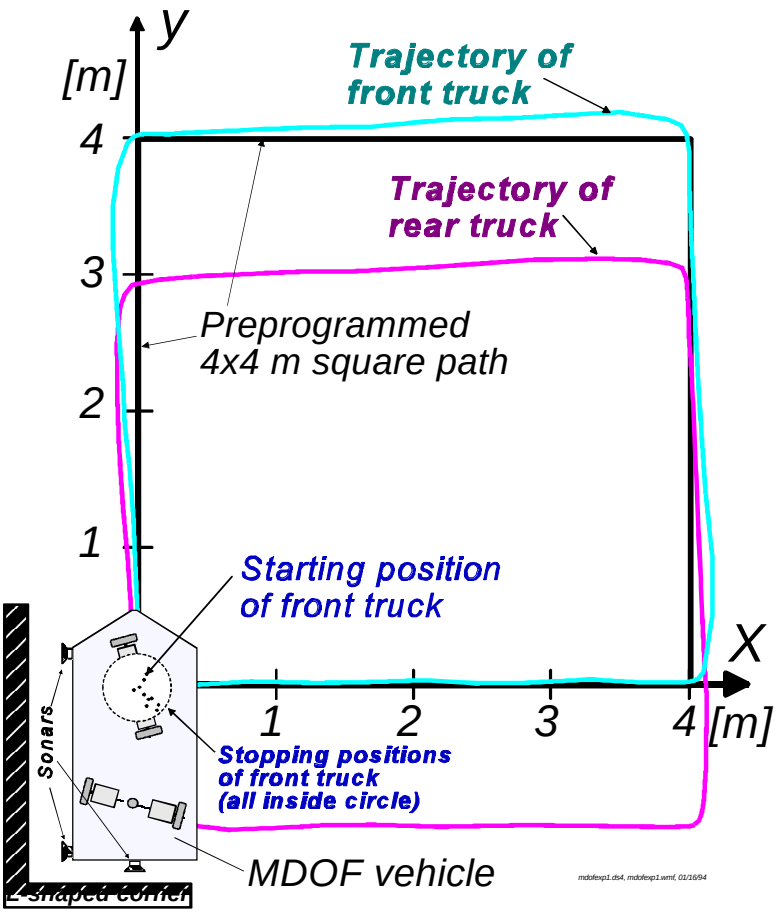
The *compliant linkage* design does not overcome errors produced by any of the above listed causes. But, as the experimental results below show, *compliant linkage* reduces the position errors of an MDOF vehicle to a level comparable to that of a 2-DOF vehicle.

We report here on results from three different experiments. The *square path experiment* in Section 6.1 was selected because it can be compared to results of similar experiments that are reported in the literature [2, 12]. The r*andom path experiment* in Section 6.2 was selected to exclude the possibility of a systematic reduction of errors which might be suspected in the highly structured *square path experiment* (although we believe that the *square path experiment*, if performed in **both** clockwise and counter-clockwise direction, is a very good test for vehicle-specific errors). Section 6.3 shows the results of the *link-length experiment*.

6.1 The Square path experiment

In this experiment the vehicle was programmed to *pass by* the corners of a 4×4 m square, as shown in Fig. 12. To provide fluid, uninterrupted motion, the programmed path did not require the vehicle to stop at the intermediate points — passing-by at a distance of less than 0.2 m was sufficient. In order to measure the position errors after completing the path, the vehicle began and ended each run in an L-shaped corner, as shown in Fig. 12. Three ultrasonic sensors were mounted on the vehicle, two sensors were facing the long side of the L-shaped corner, the third sensor faced the short side. The ultrasonic sensor system allowed measurement of the absolute position of the vehicle to within ±2 millimeters in the x and y directions, and to about $\pm 0.5^{o}$ in orientation.

At the beginning of each run a sonar measurement was taken to determine the starting position of the vehicle. The vehicle then traveled through the programmed path and returned to the L-shaped corner, where the *perceived* position (i.e., the position the vehicle "thought" it had, based on dead-reckoning) was recorded. Then, a sonar measurement was taken to determine the *absolute* position. The difference between the absolute position and the perceived position was the *position error*. Fig. 13 shows the position errors for 5 runs in clockwise, and 5 runs in counter-clockwise direction. The worst error in all ten runs was $\sqrt{7.2^2 + 8^2} = 10.7$ cm in translation; all errors were less than $\pm 1^o$ in orientation. The average distance traveled in each run was approximately $L = 15.5$ m (slightly less than 4×4 m since the vehicle didn't have to go precisely through the pre-programmed corner points), and the average speed in each run was slightly below 0.5 m/sec.



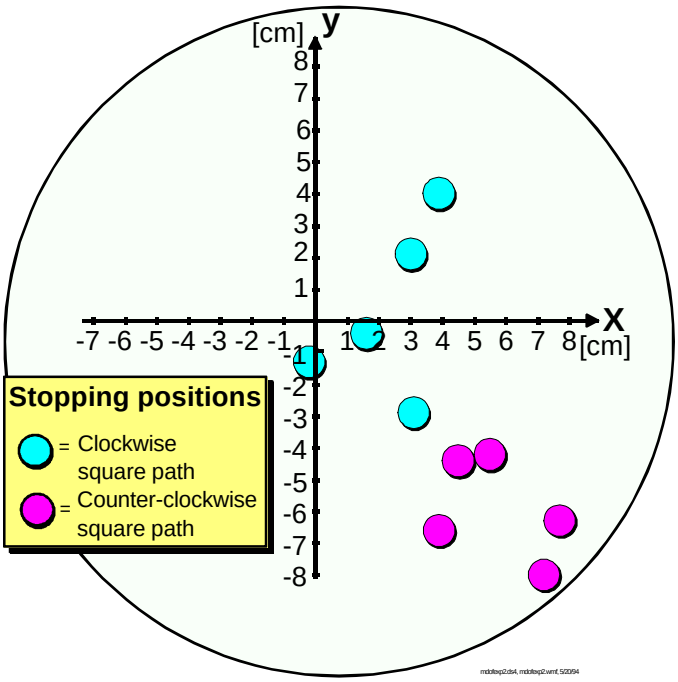
(32) **Figure 12:** The 4×4 m square path experiment.

These experimental results compare well with experimental results obtained by the author in earlier work [2] with a very accurate 2-DOF mobile robot. In those experiments the maximal lateral error was 6.5 cm and the orientation errors were less than $\pm 1^o$. The experiment with the 2-DOF vehicle differed in that the path was 50% shorter (a 2×2 m square) and the vehicle *stopped* at the corner points. The sales literature of the commercially available 2-DOF *synchro-drive* vehicle from [23] quotes an accuracy of 95.5% for a similar experiment, which, however, was apparently not run in **both** cw and ccw directions. Our worst result of 10.7 cm error corresponds to 99.3% accuracy.

6.2 The random path experiment

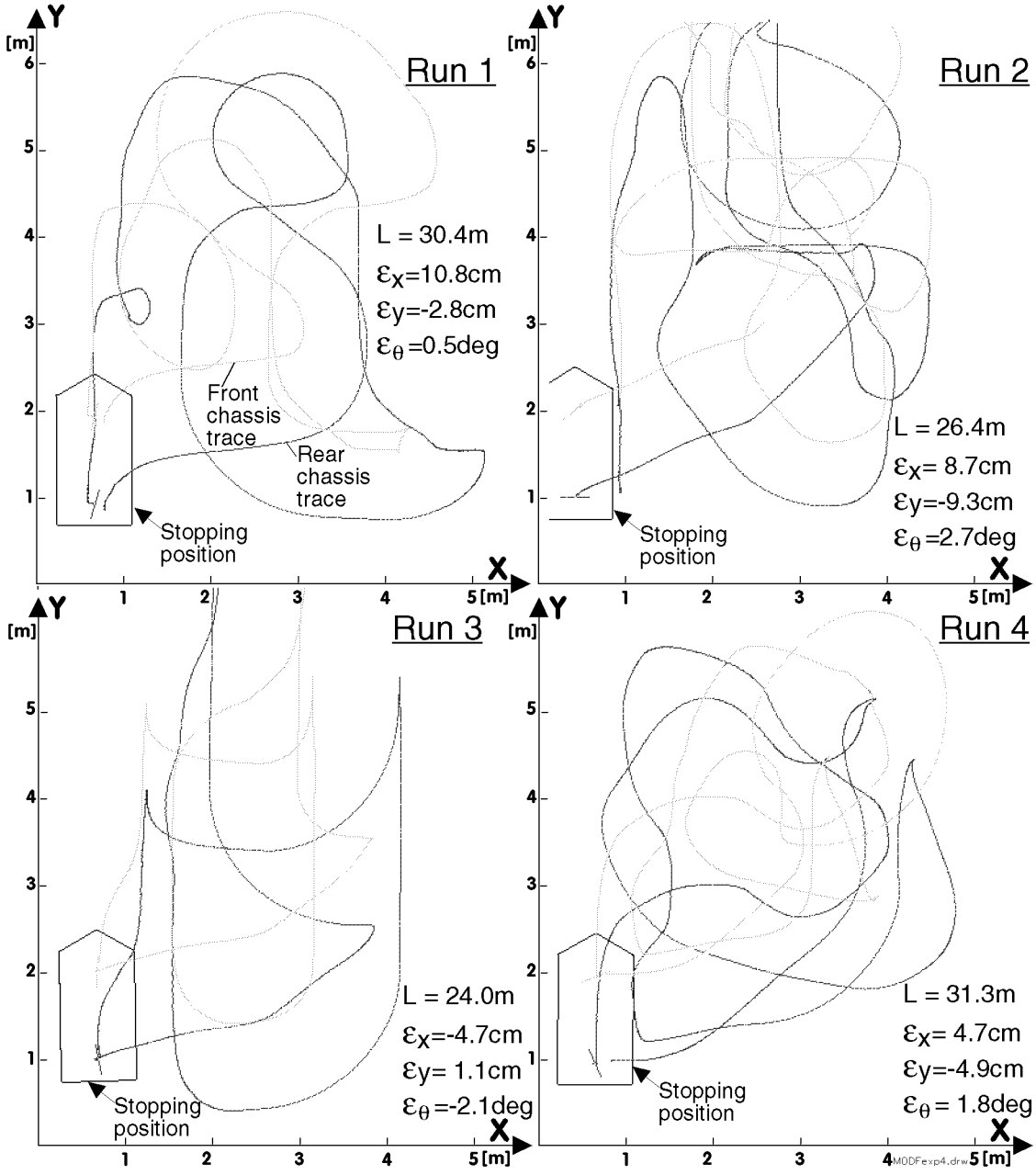
While the square path experiment was highly structured to allow comparison with other vehicles, the random path experiment is particularly unstructured, to obtain information on the behavior of the vehicle under real-world conditions. Fig. 14 shows four different experiments in which the vehicle was controlled by joystick. Each one of the four graphs shows the trace of the front and

rear truck, as well as the position of the vehicle after completing the path. The overall travel distance L is shown for each run, and so are the lateral errors ϵ_x, ϵ_y, and the orientation error ϵ_q. The errors were determined by comparing dead-reckoning information with sonar position measurements after completing a run, as explained in Section 6.1. The average speed for these runs was between 0.4 and 0.5 m/sec.



(32)**Figure 13:** Position errors after completing the 4×4 m square path of Fig. 12. Five runs proceeded in clockwise, and five runs in counter-clockwise direction.

In all four runs in Fig. 14 crabbing motion and steering (i.e., changes in orientation) were mixed and were performed simultaneously. Under joystick control, we tried to create "wild" and unpredictable movements, with many loops and turns. In Run 3, we emphasized quick directional changes; for example, from 90°-sideways crabbing right to 90°-sideways crabbing left. Although the traces look quite confusing, it is easy to distinguish between mostly sideways-crabbing motion where the traces of the front and rear truck are further apart and mostly aligned motion, where the traces are closer together. It should be noted that the results shown in Fig. 14 are representative for the results from a much larger number of similar experiments. In these experiments the translational error was consistently less than 0.5% of the total path length. These results compare favorably with the motion accuracy reported by Killough and Pin [15] for a vehicle using an "orthogonal-wheels" concept for omnidirectional motion. Killough and Pin's platform was reported to have an error of less than 1% when motion was omnidirectional but only translational (like in our square path experiment), and errors of up to 10% of the total rotation were reported when translational *and* rotational motion was performed.



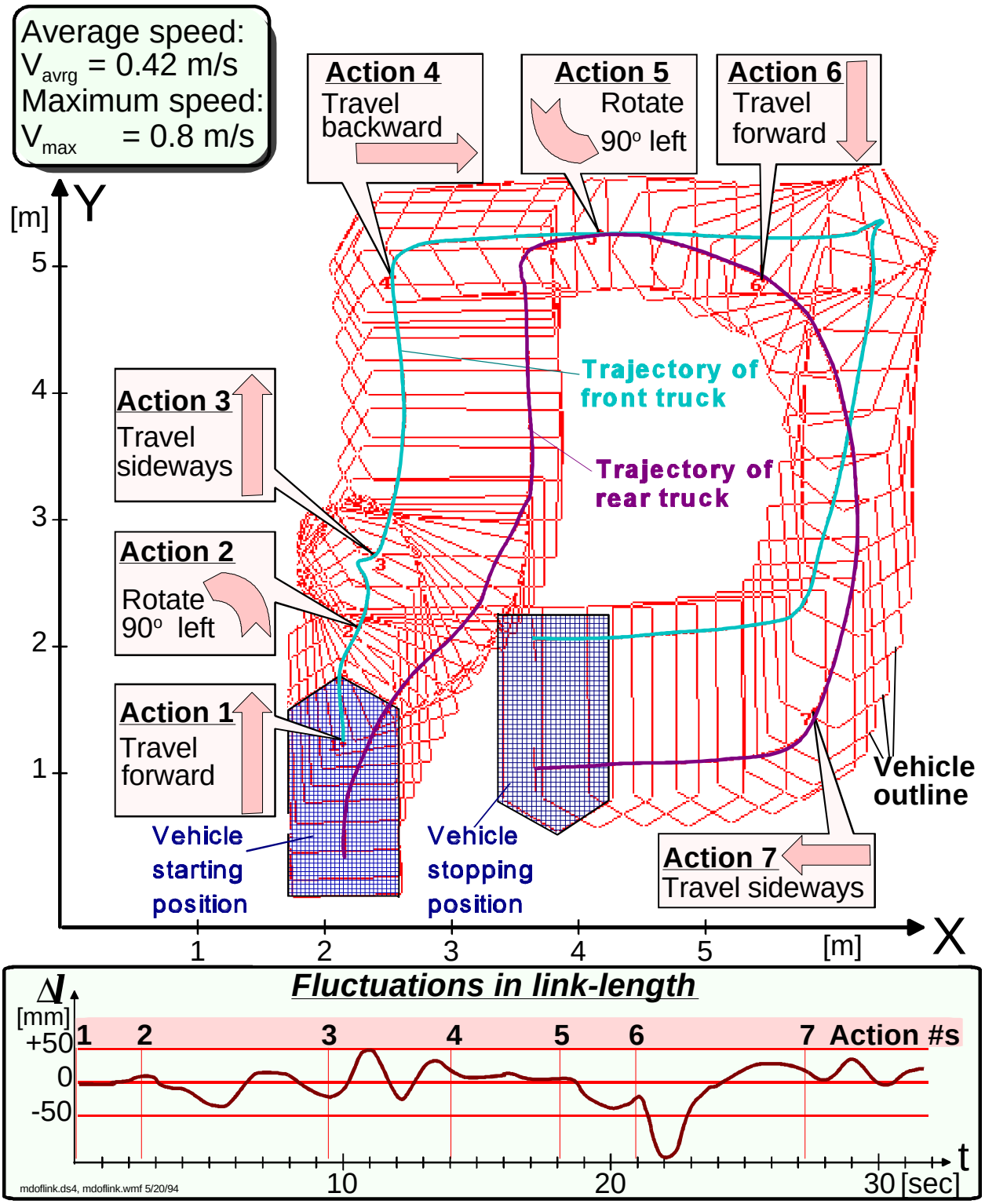
(32)**Figure 14:** Traces of the front and rear trucks under vigorous joystick control. ϵ_x, ϵ_y, and ϵ_θ show the errors after each run.

6.3 The link-length experiment

The experiment described in this Section documents the function of the controller system, specifically of the vehicle level controller. This controller aims at minimizing the *fluctuation* of the length of the compliant link, Δl: it is desirable that Δl remains small (relative to the vehicle size) since large fluctuations would be difficult to accommodate from an engineering point of view.

Figure 15a shows "snapshots" of the experimental vehicle during the execution of a preprogrammed motion-sequence. The trajectories of the center points of the front and rear trucks are also plotted. Seven different motions (labeled "Action 1" through "Action 7" in Fig. 15a) were performed, and the location of the front truck — at the moment a new Action was invoked — is marked. The motions include forward and backward travel, rotation, and sideways crabbing, as well as the combination of these components. Furthermore, the whole sequence was performed fluently, without stopping between Actions. The maximum speed was set to V_{max} = 0.8 m/s. However, the trajectory interpolator reduces the maximum speed temporarily as a function of the rate of directional changes of the individual trucks. Consequently, the average speed for the run was V_{avrg} = 0.42 m/s.

Figure 15b shows the fluctuations in link-length during the run. The maximum deviation from



(32)

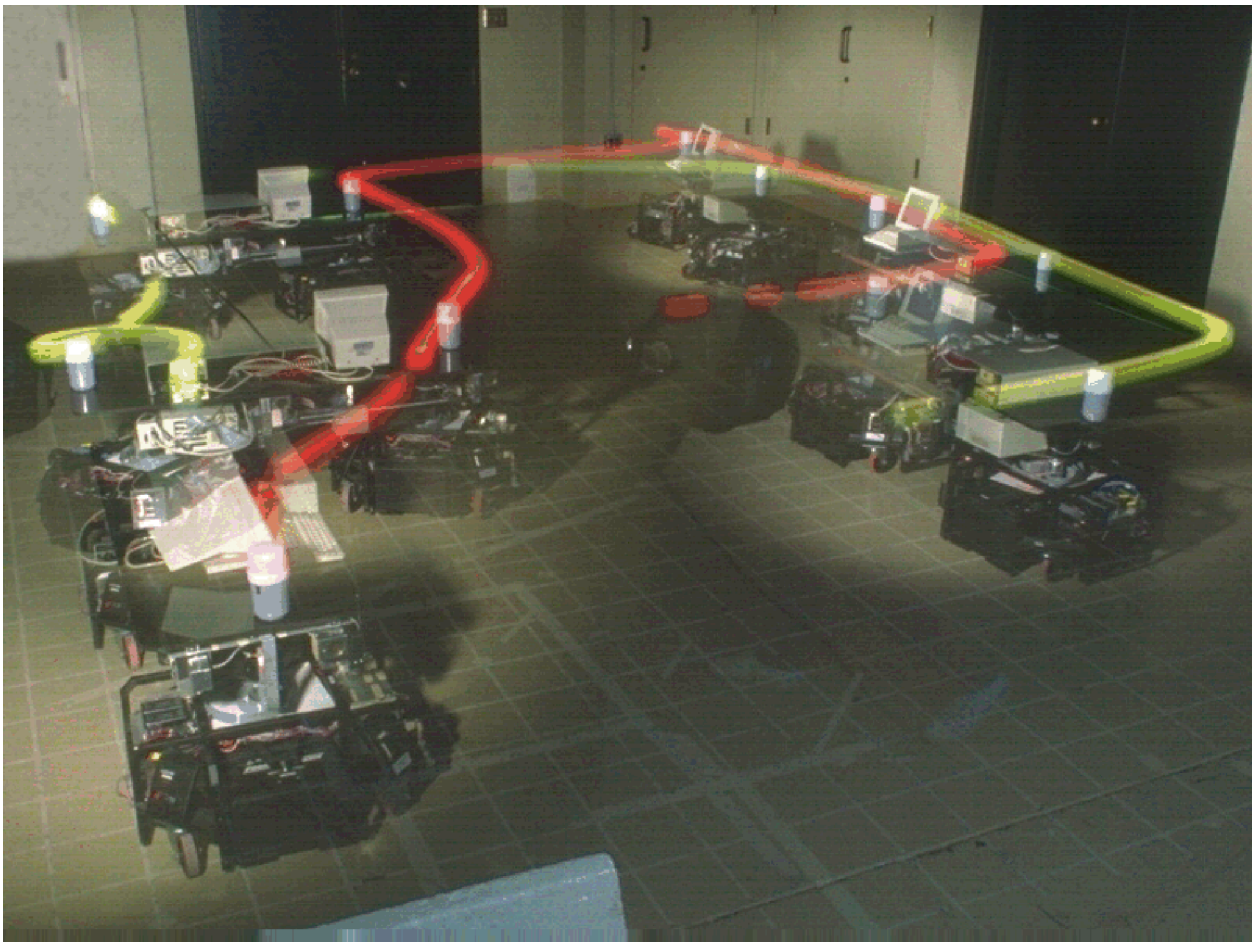
Figure 15: The link-length experiment.

the nominal link-length L was approximately Δl = 12 cm, and it occurred shortly after Action 6 was invoked. The compliant link shortened at this point because the rear truck came to a complete halt to change direction (in the upper-right corner of Fig. 15a) while the front truck was further approaching.

Figure 16 shows a multiple-exposure photograph of our experimental vehicle during the link-length experiment. One can compare the trajectories of Fig. 15 and Fig. 16. A 3-minute video segment showing U of M's experimental 4-DOF vehicle in motion is included in the *Video Proceedings* of the 1994 IEEE International Conference on Robotics and Automation [7].

8. CONCLUSIONS

We have introduced a new design for four-degree-of-freedom (4DOF) mobile robots. In this design a *compliant linkage* is used to accommodate temporary controller errors, which would otherwise violate the "rigid body constraint" and consequently cause wheel slippage.



(32) **Figure 16:** Multiple exposure photography of Michigan's 4-DOF vehicle during the link-length experiment. Light sources on the front and rear chassis leave visible traces of the the vehicle's path.
Photo: David Kother

An experimental vehicle was built and extensively tested. The multi-level control system was found to provide reliably smooth and stable motion at speeds of up to 0.5 m/s — even under vigorous joystick control. We believe that much higher speeds are feasible with more adequately designed drive trucks (in our experimental vehicle we used off-the-shelf platforms, each with two heavy 12-Volt batteries onboard, which dramatically limited the responsiveness of the truck to steering commands). We should also point out that the control law and the control parameters expressed in Equations (18) - (20) are strictly empirical. A rigorous mathematical analysis and formal design of a controller is likely to provide even better results.

The concept of *compliant linkage* provides substantially improved dead-reckoning accuracy over existing 4-DOF vehicles, and is therefore of great advantage for the operation of autonomous or semi-autonomous multi-degree-of-freedom vehicles.

In our future work we will further investigate the possibility of real-time adaptive self-calibration of the vehicle during operation, to improve dead-reckoning accuracy even beyond the accuracy achieved with conventional 2-DOF vehicles. The potential feasibility of such an approach stems from the fact that our *dual differential drive* vehicle has very accurate encoders that measure the rotation and position of the two trucks relative to the vehicle body. Under certain operating conditions, these measurements can serve as absolute measurements with respect to an external frame. The measurements can then be used to adaptively calibrate the wheel encoders (in real-time) such as to compensate for systematic external errors, caused by unequal wheel diameters or non-point wheel contact.

One reviewer suggested to overcome the problem of 4DOF mobile robots by using *torque control* instead of the conventionally used velocity control. We have not tested the torque control approach, and we are not aware of any results published on torque control for 4DOF mobile robots. While torque control may *reduce* the excessive slippage caused by violations of the rigid body constraint, it cannot guarantee to *eliminate* all excessive slippage as the *compliant linkage* does. We believe that velocity control is also advantageous because it allows for better control of the vehicle's trajectory.

Acknowledgements:
Special thanks to Photographer David Kother for his multiple-exposure artwork in Fig. 16. Special thanks also to the reviewers of this paper; their many valuable comments and suggestions were most welcome and instructive.
This research was funded by NSF grant # DDM-9114394 and in part by the Department of Energy.

---

9. BIBLIOGRAPHY

1. Alexander, J.C. and Maddocks, J.H., 1989, "On the Kinematics of Wheeled Mobile Robots." *Int. Journal of Robotics Research*, Vol. 8, No. 5, August, pp. 15-27.

2. Borenstein, J. and Koren, Y., 1987, "Motion Control Analysis of a Mobile Robot." *Transactions of ASME, Journal of Dynamics, Measurement and Control*, Vol. 109, No. 2, pp. 73-79.
3. Borenstein, J. and Raschke, U., 1991, "Real-time Obstacle Avoidance for Non-Point Mobile Robots. *Proceedings of the Fourth World Conference on Robotics Research*, Pittsburgh, Pennsylvania, Sept. 17-19, 1991, pp. 2.1 - 2.9.
4. Borenstein, J., 1992, "Compliant-linkage Kinematic Design for Multi-degree-of-freedom Mobile Robots ." *Proceedings of the SPIE Symposium on Advances in Intelligent Systems, Mobile Robots VII*, Boston, MA, Nov. 15-20, 1992, pp. 344-351.
5. Borenstein, J., 1993, "Multi-layered Control of a Four-Degree-of-Freedom Mobile Robot With Compliant Linkage." *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, May 2-7, pp. 3.7 - 3.12.
6. Borenstein, J., 1994, "The CLAPPER: a Dual-Drive Mobile Robot with Internal Correction of Dead-reckoning Errors." *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, May 8-13, 1994, pp. 3085-3090.
7. Borenstein, J., 1994Video, "Four-Degree-of-Freedom Redundant Drive Vehicle With Compliant Linkage." ***Video Proceedings*** *of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, May 8-13, 1994. (Note: Video Proceedings booklet lists this entry under incorrect title and abstract).
8. DOE-1990, Office of Environmental Restoration and Waste Management, Office of Technology Development, "Environmental Restoration and Waste Management) Robotics Technology Development Program, Robotics 5-Year Program Plan." Program Plan, DOE/CE-0007T, Vol. II, p. 50.
9. DOE-1991, Office of Energy Research, "Solicitation for 1992 SBIRs", DOE/ER-0509, p. 58.
10. Evans et al, Pat. # 4,932,489, 1990, "Steering and Drive Means for Robot Vehicle."
11. Feng, D., Friedman, M. B., and Krogh, B. H., 1989, "The Servo-Control System for an Omnidirectional Mobile Robot." *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, Scottsdale, Arizona, May 14-19, 1989, pp. 1566-1571.
12. Feng, L, Koren, Y., and Borenstein, J., 1993, "A Cross-Coupling Motion Controller for Mobile Robots." *IEEE Journal of Control Systems Magazine*, December 1993, pp. 35-43.
13. Hammond, G., 1986, "*AGVs at work: Automated Guided Vehicle Systems*." Bedford, New York.
14. Killough, S. M. and Pin, F. G., 1990, "A Fully Omnidirectional Wheeled Assembly for Robotic Vehicles." *Transactions of the American Nuclear Society 1990 Annual Meeting*, Nashville, Tennessee, June 10-14.
15. Killough, S. M. and Pin, F. G., 1992, "Design of an Omnidirectional Holonomic Wheeled Platform Prototype." *Proceedings of the IEEE Conference on Robotics and Automation*, Nice, France, May 1992, pp. 84-90.
16. Leifer, L.J., Van der Loos, H.F. Chalowski, S.J., 1988, "Development of an Omni-directional Mobile Vocational Assistant Robot," *Proceedings of the International Conference of the Association for the Advancement of Rehabilitation Technology*, Montreal, Canada.
17. Moravec, H. P., 1984, "Three Degrees for a Mobile Robot." Carnegie-Mellon University, *The Robotics Institute*, Mobile Robots Laboratory, Technical Report.

18. Pritschow, G., Jantzer, M., and Schumacher, H., 1988, "A Control System for Free-ranging Mobile Robots In Production Lines." *Proceedings of the 6th International Conference on Automated Guided Vehicle Systems 1988*, Brussels, Belgium, 25-26 October, pp. 189-197.
19. Reister, D. B. et al., 1991, "DEMO 89) The Initial Experiment With the HERMIES-III Robot." *Proceedings of the IEEE Conference on Robotics and Automation* Sacramento, California, April 7-12, , pp. 2562-2567.
20. Reister, D. B., 1991, "A New Wheel Control System for the Omnidirectional HERMIES-III Robot." *Proceedings of the IEEE Conference on Robotics and Automation* Sacramento, California, April 7-12, pp. 2322-2327.
21. Reister, D.B. and Unseren, M.A., 1993, "Position and Constraint Force Control of a Vehicle with Two or More Steerable Drive Wheels." *IEEE Transactions on Robotics and Automation*.Vol. 9, No. 6, December, pp. 723-731.
22. Wiklund, U., Andersson, U., and Hyyppä, K., 1988, "AGV Navigation by Angle Measurement." *Proceedings of the 6th International Conference on Automated Guided Vehicle Systems 1988*, Brussels, Belgium, 25-26 October, pp. 199-212.

Commercial References:

23. Cybermotion, 5457 JAE Valley Road, Roanoke, Virginia 24014.
24. Hewlett Packard, "Optoelectronics Designer's Catalog, 1991-1992."
25. Transition Research Corporation (TRC), Shelter Rock Lane, Danbury, Connecticut 06810.