2010 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
MODELING & SIMULATION, TESTING AND VALIDATION (MSTV) MINI-SYMPOSIUM
AUGUST 17-19 DEARBORN, MICHIGAN

KINEMATIC ANALYSIS FOR AUTOMATIC PLOWING DEPTH CONTROL OF A MINE CLEARING COMBAT VEHICLE

Fuh-Feng Tsai, PhD
Kevin Strittmatter, PE
BAE Systems, U.S. Combat Systems
York, PA

ABSTRACT

This paper develops a linear closed form equation as required for automatic plowing depth control of a mine clearing combat vehicle. The vehicle will be tasked with using its Mine Clearing Blade (MCB) to remove surface laid and buried land mines on undulating terrain so that other vehicles can follow its path without the threat of mines. Blade control must be automatic to ensure that the target depth of the cleared path is achieved and all mines on the path are removed. A closed form solution for real-time computing relating the MCB motion and hydraulic actuator movement is developed and implemented. The equations are provided in symbolic form so that the dimensions of the mechanism can be directly substituted and/or modified without re-derivation. Results were verified with field measured data and implemented in the controller of a real vehicle to successfully achieve objective goal of Automatic Mine Clearing.

INTRODUCTION

An automatic plowing depth control requires the resulting controller integrates kinematics, soil mechanics, control algorithms, hydraulic power, and vehicle dynamics. Owing to the interactive nature of the system, the control software must be efficient and run in real-time with a constant step size equal to the sample period of control system. Commercial software such as DADS / VL Motion or ADAMS are widely used for off-line simulation but are infeasible due to computational speed, availability of source or binary code for embedded targets, integration step size control for convergence does not meet real-time application, non-linear algebraic joint or driving constraint equations may require many iterations for convergence to satisfy specified error tolerance that require CPU time beyond the sampling period in a control loop, etc. Therefore a closed form solution for real-time computing relating the MCB motion and hydraulic actuator movement is developed and implemented.

GOALS AND ASSUMPTIONS

The equations developed in this paper are for Real-Time operation that has different goals compared with those of general purpose codes such as ADAMS or DADS as follows,

1. It is intent to reduce depth plow error by providing good (not exact) kinematic relationship between blade geometry and hydraulic actuators with a minimum set of generalized coordinates. Error is to be corrected via control loop with sensor feedbacks in real applications at the next sample step rather than current time step. Computation must be completed within one sample period. The goal is to reduce error and to meet real-time application. Commercial codes use a larger set of generalized coordinates so the kinematic relationship can be more precisely formulated. However, computation is intensive since constraints must be satisfied with a specified tight error tolerance via iterations and in general can’t be completed within a sample real time step. The goal is for accuracy but not Real-Time application.
2. For joint constraints, computer models require iterations (e.g., Newton-Raphson) to check non-linear constraint equations for initial assembly and joint separation. The physical blade in Real-Time operation is assumed well assembled and its joints are not broken or deformed during operation; therefore, iterations for checking joint separations are not required.
3. For driving constraints, computer models use iterations to check nonlinear constraints. In Real-Time application, driving constraints are not explicitly implemented so iterations are not carried out. A physical hydraulic system is controlled by pressure

(force) and flow (speed). When a servo control valve is opened by a current command, the actuator is trying to move with a desired speed (by flow) rather than a desired length. Linear transducers are used to measure the length for error correction in the next time step. Therefore, the goal is to develop velocity relationship for valve command rather than imposing position or velocity driving constraints as used in the off-line simulations. Variation method is used to meet this goal by developing the perturbation ratio between blade and actuator movements. Given a sample frequency of control loops, the perturbation ratio is assumed to be a constant speed ratio within the time limit of the sample period at a given blade position.

4. During plowing, the effective mass, in addition to the original blade weight, is increased with soil weight and plow force. Since power (Pressure/Force or Flow/Speed) of hydraulic system is always limited, the natural frequency of blade system is reduced and system response to command slows down. Trying to move the blade to satisfy driving constraints within the limited sample period requires significant power that is not feasible if not impossible. Therefore for Real-Time application, achieving stable plow control within an acceptable error tolerance for a long duration operation is more important than to strictly meet kinematic constrains via iterations in every time step. The later might diverge due to computation time limit within sample period and cause unstable control.

MOTION DESCRIPTION

The kinematic relationships between the Mine Clearing Blade (MCB) and the chassis, as shown in figure 1, are represented by six nonlinear constraint equations -- three joint constraints and three driving constraints. The three joint constraints are not controllable and must be satisfied no matter how the hydraulic system performs. They can be stated as follows:

1. The length of left push beam is constant and can't be changed with time.
2. The length of right push beam is constant and can't be changed with time.
3. The length of the diagonal brace between the left push beam and the chassis can't be changed with time, so that there is no side-to-side swing motion between the left push beam and the chassis.

The remaining three driving constraints are controllable by changing the length of the three actuators which control the roll, pitch, and lift motion of the blade. Combining the three joint constraints with the three driving constraints, the six generalized coordinates $\mathbf{q} = [x, y, z, \phi, \theta, \varphi]^T$ of the MCB relative to the chassis frame are determined, i.e., the motion is kinematically driven.

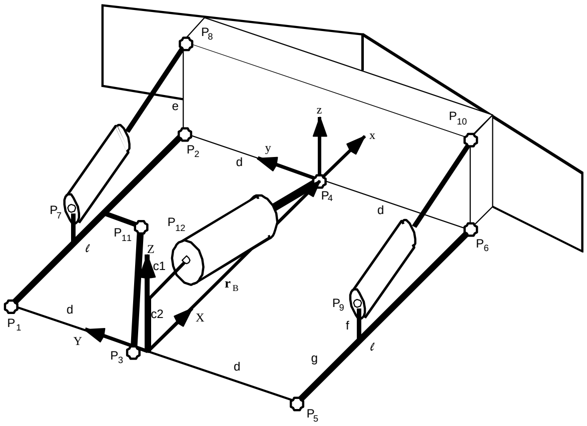

Figure 1: Mine Clearing Blade of the Combat Vehicle

KINEMATIC ANALYSIS

Since the relative motion between the blade and the chassis is our concern, all equations are derived in the chassis frame rather than the earth frame to avoid unnecessary transformations. The transformation matrix from the blade frame to the chassis frame is simply denoted by "$\mathbf{A}$ ". Therefore, a constant vector on the blade frame can be represented in the chassis frame [3] as $\mathbf{s} = \mathbf{A}\mathbf{s}'$

Refer to figure 1, the frames X-Y-Z and x-y-z represent the chassis and the blade frames, respectively. Points P_j (j=1..., 12) denotes the joint attachment points. The system consists of two rigid bodies, the chassis and the MCB, and its relative motion must be described with the generalized coordinates $\mathbf{q} = [x, y, z, \phi, \theta, \varphi]^T$. The two tilt actuators are attached to the push beams that are modeled as distance linkages without introducing additional generalized coordinates. Therefore, minor approximation is unavoidable in writing the equations for the two tilt actuators in terms of the only available generalized coordinates $\mathbf{q}$ for the blade.

Joint Constraint for the Left Push Beam

Statement: The length of left push beam is constant and can't be changed with time.

The position vectors $\mathbf{P}_1$ and $\mathbf{P}_2$ in the chassis frame are

$$\mathbf{P}_1 = \mathbf{s}_1 = [0 \quad d \quad 0]^T \tag{1}$$

and

$$\mathbf{P}_2 = \mathbf{r}_B + \mathbf{A}\mathbf{s}_2' \tag{2}$$

respectively, where

$$\mathbf{s}_2' = \begin{bmatrix} 0 & d & 0 \end{bmatrix}^T \tag{3}$$

is a constant vector from the origin of the blade reference frame to the point P_2 and is defined in the blade frame; $\mathbf{r}_B$ is the position vector from the origin of the chassis reference frame to the origin of the blade reference frame and is defined as

$$\mathbf{r}_B = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{4}$$

The relative roll, pitch, and yaw angle of the blade with respect to the chassis frame are denoted by ϕ, θ, φ respectively. Using roll-pitch-yaw rotational sequence, the transformation matrix $\mathbf{A}$ can be derived as

$$\mathbf{A} = \begin{bmatrix} c(\theta)c(\varphi) & -c(\theta)s(\varphi) & s(\theta) \\ s(\phi)s(\theta)c(\varphi)+c(\phi)s(\varphi) & -s(\phi)s(\theta)s(\varphi)+c(\phi)c(\varphi) & -s(\phi)c(\theta) \\ -c(\phi)s(\theta)c(\varphi)+s(\phi)s(\varphi) & c(\phi)s(\theta)s(\varphi)+s(\phi)c(\varphi) & c(\phi)c(\theta) \end{bmatrix}$$
$$\equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{5}$$

The position vector on the left push beam from point P_1 to P_2 is

$$\mathbf{L} = \mathbf{P}_2 - \mathbf{P}_1 = \mathbf{r}_B + \mathbf{A}\mathbf{s}_2' - \mathbf{s}_1 \tag{6}$$

and its square length is

$$\begin{aligned} \ell^2 &= (\mathbf{r}_B + \mathbf{A}\mathbf{s}_2' - \mathbf{s}_1)^T(\mathbf{r}_B + \mathbf{A}\mathbf{s}_2' - \mathbf{s}_1) \\ &= \mathbf{r}_B^T\mathbf{r}_B + \mathbf{r}_B^T\mathbf{A}\mathbf{s}_2' - \mathbf{r}_B^T\mathbf{s}_1 + \mathbf{s}_2'^T\mathbf{A}^T\mathbf{r}_B + \mathbf{s}_2'^T\mathbf{A}^T\mathbf{A}\mathbf{s}_2' - \mathbf{s}_2'^T\mathbf{A}^T\mathbf{s}_1 \\ &\quad - \mathbf{s}_1^T\mathbf{r}_B - \mathbf{s}_1^T\mathbf{A}\mathbf{s}_2' + \mathbf{s}_1^T\mathbf{s}_1 \end{aligned} \tag{7}$$

where

$$\mathbf{r}_B^T\mathbf{r}_B = x^2 + y^2 + z^2 \tag{8}$$

$$\mathbf{r}_B^T\mathbf{A}\mathbf{s}_2' = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 0 \\ d \\ 0 \end{bmatrix} = d(a_{12}x + a_{22}y + a_{32}z) \tag{9}$$

$$\mathbf{r}_B^T\mathbf{s}_1 = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} 0 \\ d \\ 0 \end{bmatrix} = dy \tag{10}$$

$$\mathbf{s}_1^T\mathbf{A}\mathbf{s}_2' = d^2 a_{22} \tag{11}$$

$$\mathbf{s}_1^T\mathbf{s}_1 = d^2 \tag{12}$$

$$\mathbf{s}_2'^T\mathbf{A}^T\mathbf{A}\mathbf{s}_2' = \mathbf{s}_2'^T\mathbf{I}\mathbf{s}_2' = \mathbf{s}_2'^T\mathbf{s}_2' = d^2 \tag{13}$$

since all terms on the right side of equation (7) are scalar,

$$\mathbf{s}_2'^T\mathbf{A}^T\mathbf{r}_B = (\mathbf{s}_2'^T\mathbf{A}^T\mathbf{r}_B)^T = \mathbf{r}_B^T\mathbf{A}\mathbf{s}_2' \tag{14}$$

$$\mathbf{s}_1^T\mathbf{r}_B = \mathbf{r}_B^T\mathbf{s}_1 \tag{15}$$

$$\mathbf{s}_2'^T\mathbf{A}^T\mathbf{s}_1 = \mathbf{s}_1^T\mathbf{A}\mathbf{s}_2' \tag{16}$$

Equation (7) can then be expanded as

$$\begin{aligned} \ell^2 &= x^2 + y^2 + z^2 + 2d(a_{12}x + a_{22}y + a_{32}z) \\ &\quad - 2dy - 2d^2a_{22} + 2d^2 \end{aligned} \tag{17}$$

Joint Constraint for the Right Push Beam

Statement: The length of right push beam is constant and can't be changed with time.

The position vectors $\mathbf{P}_5$ and $\mathbf{P}_6$ for the right push beam are defined in the chassis frame as

$$\mathbf{P}_5 = \mathbf{s}_5 = \begin{bmatrix} 0 & -d & 0 \end{bmatrix}^T \tag{18}$$

and

$$\mathbf{P}_6 = \mathbf{r}_B + \mathbf{A}\mathbf{s}_6' \tag{19}$$

respectively, where

$$\mathbf{s}_6' = \begin{bmatrix} 0 & -d & 0 \end{bmatrix}^T \tag{20}$$

Following the same derivation described above, the distance constraint for the right push beam can be obtained as

$$\begin{aligned} \ell^2 &= x^2 + y^2 + z^2 - 2d(a_{12}x + a_{22}y + a_{32}z) \\ &\quad + 2dy - 2d^2a_{22} + 2d^2 \end{aligned} \tag{21}$$

Joint Constraint for the diagonal brace

Statement: The length of the diagonal brace between the left push beam and the chassis can't be changed with time, so that there is no side-to-side swing motion between the left push beam and the chassis.

The diagonal brace prevents side-to-side swing motion of the left push beam, i.e., the position vector $\mathbf{L}$ defined from point P_1 to point P_2 is always perpendicular to the chassis Y axis. Any roll motion of the blade relative to the chassis results in positive y deviation of the origin of the blade frame, as shown in figure 2. This causes asymmetric motion of the blade because side-to-side swing motion of the right push beam must occur.

Since yaw or pitch motion of the blade does not contribute y deviation of the origin, the kinematic relationship between y and roll ϕ can be written as

$$y = d(1-\cos\phi) \tag{22}$$

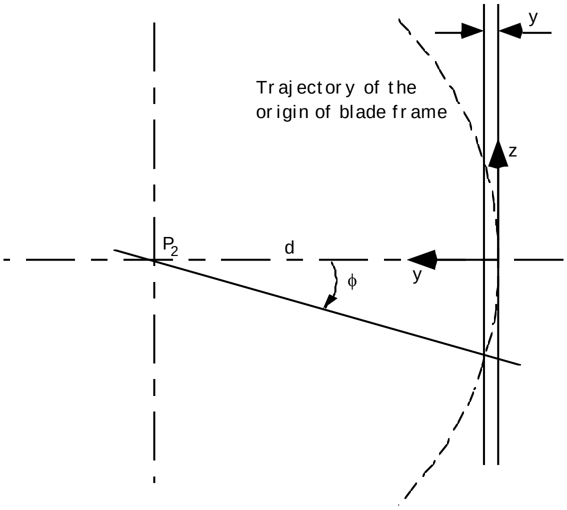

Figure 2: Roll Motion of the Blade

Redundant Constraint

In general, the joint constraints in equations (17), (21), and (22) define the relationship between the six generalized coordinates x, y, z, ϕ, θ, and φ. With three algebraic equations and six variables, the blade therefore has 3 degrees of freedom before the three actuators are attached. However, when the yaw and roll angles are zero, both equations (17) and (21) reduces to

$$\ell^2 = x^2 + z^2 \tag{23}$$

and one of the kinematic constraints is lost. Equations (17) and (21) become redundant for this specific configuration. Redundant constraints will cause the constraint Jacobian matrix to be singular or ill-conditioned and should be avoided. Therefore, an additional kinematic constraint must be imposed.

Since yaw motion of the blade is not allowed due to the unchanged lengths of two push beams, the constraint

$$\varphi = 0 \tag{24}$$

can be used to replace the redundant one. Equations 17 and 21 can be summed up to represent a general constraint as

$$x^2 + y^2 + z^2 + 2d^2[1 - c(\phi)] = \ell^2 \tag{25}$$

With the joint constraint in equation (24) imposed, the system is reduced to contain 5 generalized coordinates $\mathbf{q} = [x, y, z, \phi, \theta]^T$ and two joint constraints, as shown in equations (22) and (25). Note that the two constraints do not contain the variable θ. It means the pitch motion must be controlled by actuators and will not be restricted by the joint constraints. The transformation matrix $\mathbf{A}$ is thus reduced to

$$\mathbf{A} \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ s(\phi)s(\theta) & c(\phi) & -s(\phi)c(\theta) \\ -c(\phi)s(\theta) & s(\phi) & c(\phi)c(\theta) \end{bmatrix} \tag{26}$$

Adjustment of Generalized Coordinates

The five generalized coordinates $\mathbf{q} = [x, y, z, \phi, \theta]^T$ are not independent and should be adjusted to satisfy the joint constraints. Equations (22) and (25) can be rewritten as

$$\Phi^1 = y - d + dc(\phi) = 0 \tag{27}$$

and

$$\Phi^2 = x^2 + y^2 + z^2 - 2d^2c(\phi) + 2d^2 - \ell^2 = 0 \tag{28}$$

respectively, where the superscript of Φ denotes the constraint equation number..

Taking the variation of equations (27-28) with respect to generalized coordinates and writing in a compact form yields

$$\boldsymbol{\Phi}_{\mathbf{q}}^{J}\delta\mathbf{q} = \begin{bmatrix} \Phi_{\mathbf{q}}^{1} \\ \Phi_{\mathbf{q}}^{2} \end{bmatrix}\delta\mathbf{q} = \begin{bmatrix} 0 & 1 & 0 & -ds(\phi) & 0 \\ x & y & z & d^2s(\phi) & 0 \end{bmatrix}\begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta\phi \\ \delta\theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{29}$$

where

$$\boldsymbol{\Phi}_{\mathbf{q}}^{J} = \begin{bmatrix} 0 & 1 & 0 & -ds(\phi) & 0 \\ x & y & z & d^2s(\phi) & 0 \end{bmatrix} \tag{30}$$

is the Jacobian matrix associated with the two joint constraints, and

$$\delta\mathbf{q} = [\delta x \quad \delta y \quad \delta z \quad \delta\phi \quad \delta\theta]^T \tag{31}$$

is the variation of generalized coordinates.

Taking the time derivatives of equations (27-28) yields the velocity constraints as

$$\mathbf{\Phi}_{\mathbf{q}}^{J}\dot{\mathbf{q}} = \begin{bmatrix} 0 & 1 & 0 & -ds(\phi) & 0 \\ x & y & z & d^2s(\phi) & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{32}$$

or

$$\dot{y} - ds(\phi)\dot{\phi} = 0 \tag{33}$$

and

$$x\dot{x} + y\dot{y} + z\dot{z} + d^2s(\phi)\dot{\phi} = 0 \tag{34}$$

During the operation of the blade, the variables y, z, and ϕ might become zero, however, x is always greater than zero. Therefore, the two rows of the Jacobian are linear independent, i.e., the two constraints are not redundant all the time. The nonzero coefficients of Jacobian in the first row are corresponding to δy while the second row is corresponding to δx. It means that y and x should be determined by these two joint constraints and the remaining variables z, ϕ, and θ should be controlled by the three actuators.

Algorithm: Given the five generalized coordinates and their time derivatives, the coordinates x, y, and velocities $\dot{x}$ and $\dot{y}$ should be adjusted as follows:

$$y = d(1 - \cos\phi) \tag{35}$$

$$x = +\sqrt{\ell^2 - y^2 - z^2 - 2d^2[1 - c(\phi)]} \tag{36}$$

$$\dot{y} = ds(\phi)\dot{\phi} \tag{37}$$

$$\dot{x} = \frac{-1}{x}(y\dot{y} + z\dot{z} + d^2s(\phi)\dot{\phi}) \tag{38}$$

After the adjustment of generalized coordinates, the set $\mathbf{q} = [x, y, z, \phi, \theta]^T$ is a feasible solution that satisfies the two joint constraints. The lengths and their time derivatives of actuators can therefore be obtained in terms of $\mathbf{q}$ and $\dot{\mathbf{q}}$ as described in the following sections.

Driving Constraint and Its Jacobian Matrix for the Left Actuator

The attachment point of the left actuator on the left push beam can be represented by a position vector $\mathbf{P}_7$, as shown in figure 3, as

$$\mathbf{P}_7 = \mathbf{P}_1 + \frac{g}{\ell}\begin{bmatrix} x_2 \\ 0 \\ z_2 \end{bmatrix} + \frac{f}{\ell}\begin{bmatrix} -z_2 \\ 0 \\ x_2 \end{bmatrix} = \begin{bmatrix} \frac{g}{\ell}x - \frac{f}{\ell}[z + s(\phi)d] \\ d \\ \frac{g}{\ell}[z + s(\phi)d] + \frac{f}{\ell}x \end{bmatrix} \tag{39}$$

and the attachment point $\mathbf{P}_8$ on the blade, as shown in figure 1 is

$$\mathbf{P}_8 = \mathbf{r}_B + \mathbf{A}\mathbf{s}'_8 \tag{40}$$

where $\mathbf{s}'_8$ is a constant vector as

$$\mathbf{s}'_8 = [0 \quad d \quad e]^T \tag{41}$$

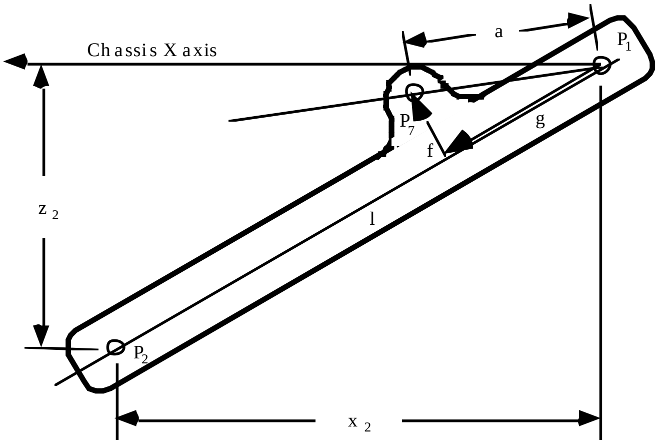

Figure 3: Left Actuator and Push Beam Attachment Points

The length vector $\mathbf{L}_\ell$ can be computed as

$$\mathbf{L}_\ell = \mathbf{P}_8 - \mathbf{P}_7 = \mathbf{r}_B + \mathbf{A}\mathbf{s}'_8 - \mathbf{P}_7 \tag{42}$$

and its square length L_ℓ is

$$
\begin{aligned}
\mathrm{L}_\ell^2 &= \mathbf{L}_\ell^{\mathrm{T}}\mathbf{L}_\ell = \mathbf{r}_{\mathrm{B}}^{\mathrm{T}}\mathbf{r}_{\mathrm{B}} + 2\mathbf{r}_{\mathrm{B}}^{\mathrm{T}}\mathbf{A}\mathbf{s}_8' - 2\mathbf{r}_{\mathrm{B}}^{\mathrm{T}}\mathbf{P}_7 + \mathbf{s}_8'^{\mathrm{T}}\mathbf{s}_8' - 2\mathbf{P}_7^{\mathrm{T}}\mathbf{A}\mathbf{s}_8' + \mathbf{P}_7^{\mathrm{T}}\mathbf{P}_7 \\
&= x^2 + y^2 + z^2 \\
&\quad + 2\{[s(\theta)e]x + [c(\phi)d - s(\phi)c(\theta)e]y + [s(\phi)d + c(\phi)c(\theta)e]z\} \\
&\quad - 2\left\{x\left[\frac{g}{\ell}x - \frac{f}{\ell}[z + s(\phi)d]\right] + yd + z\left[\frac{g}{\ell}[z + s(\phi)d] + \frac{f}{\ell}x\right]\right\} \\
&\quad - 2\left\{\left[\frac{g}{\ell}x - \frac{f}{\ell}[z + s(\phi)d]\right]s(\theta)e + [c(\phi)d - s(\phi)c(\theta)e]d \right. \\
&\quad \left. + [s(\phi)d + c(\phi)c(\theta)e]\left[\frac{g}{\ell}[z + s(\phi)d] + \frac{f}{\ell}x\right]\right\} \\
&\quad + (d^2 + e^2) + (a^2 + d^2)
\end{aligned}
\tag{43}
$$

Taking the time derivative of equation (43) yields

$$
\begin{aligned}
\mathrm{L}_\ell\dot{\mathrm{L}}_\ell &= x\dot{x} + y\dot{y} + z\dot{z} + \{[s(\theta)e]\dot{x} + [c(\theta)e\dot{\theta}]x \\
&\quad + [-s(\phi)\dot{\phi}d - c(\phi)c(\theta)e\dot{\phi} + s(\phi)s(\theta)\dot{\theta}e]y + [c(\phi)d - s(\phi)c(\theta)e]\dot{y} \\
&\quad + [c(\phi)d\dot{\phi} - s(\phi)c(\theta)\dot{\phi}e - c(\phi)s(\theta)\dot{\theta}e]z + [s(\phi)d + c(\phi)c(\theta)e]\dot{z}\} \\
&\quad - \left\{\dot{x}\left[\frac{g}{\ell}x - \frac{f}{\ell}[z + s(\phi)d]\right] + x\left[\frac{g}{\ell}\dot{x} - \frac{f}{\ell}[\dot{z} + c(\phi)\dot{\phi}d]\right]\right. \\
&\quad \left. + \dot{y}d + \dot{z}\left[\frac{g}{\ell}[z + s(\phi)d] + \frac{f}{\ell}x\right] + z\left[\frac{g}{\ell}[\dot{z} + c(\phi)\dot{\phi}d] + \frac{f}{\ell}\dot{x}\right]\right\} \\
&\quad - \left\{\left[\frac{g}{\ell}\dot{x} - \frac{f}{\ell}[\dot{z} + c(\phi)\dot{\phi}d]\right]s(\theta)e + \left[\frac{g}{\ell}x - \frac{f}{\ell}[z + s(\phi)d]\right]c(\theta)\dot{\theta}e\right. \\
&\quad + d[-s(\phi)\dot{\phi}d - c(\phi)c(\theta)\dot{\phi}e + s(\phi)s(\theta)\dot{\theta}e] \\
&\quad + [c(\phi)\dot{\phi}d - s(\phi)\dot{\phi}c(\theta)e - c(\phi)s(\theta)\dot{\theta}e]\left[\frac{g}{\ell}[z + s(\phi)d] + \frac{f}{\ell}x\right] \\
&\quad \left. + [s(\phi)d + c(\phi)c(\theta)e]\left[\frac{g}{\ell}[\dot{z} + c(\phi)\dot{\phi}d] + \frac{f}{\ell}\dot{x}\right]\right\}
\end{aligned}
\tag{44}
$$

The desired length L_ℓ and velocity $\dot{\mathrm{L}}_\ell$ of the left actuator can be obtained from equations (43-44), respectively.

A perturbation method is used here to determine the relationship between the lengths of the actuators and the generalized coordinates of the MCB, therefore, the Jacobian matrix associated with the left actuator should be constructed. The length of the left actuator L_ℓ is a function of the generalized coordinates x, y, z, ϕ, and θ, as shown in equation (43), and can be represented as

$$
\mathrm{L}_\ell = \mathrm{L}_\ell(x, y, z, \phi, \theta) \tag{45}
$$

where yaw φ is zero all the time.

Taking the variation of equation (45) yields

$$
\begin{aligned}
\delta\mathrm{L}_\ell &= \frac{\partial \mathrm{L}_\ell}{\partial x}\delta x + \frac{\partial \mathrm{L}_\ell}{\partial y}\delta y + \frac{\partial \mathrm{L}_\ell}{\partial z}\delta z + \frac{\partial \mathrm{L}_\ell}{\partial \phi}\delta\phi + \frac{\partial \mathrm{L}_\ell}{\partial \theta}\delta\theta \\
&= \begin{bmatrix} \dfrac{\partial \mathrm{L}_\ell}{\partial x} & \dfrac{\partial \mathrm{L}_\ell}{\partial y} & \dfrac{\partial \mathrm{L}_\ell}{\partial z} & \dfrac{\partial \mathrm{L}_\ell}{\partial \phi} & \dfrac{\partial \mathrm{L}_\ell}{\partial \theta} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta\phi \\ \delta\theta \end{bmatrix} \\
&= \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{L}_\ell}\delta\mathbf{q}
\end{aligned}
\tag{46}
$$

where

$$
\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{L}_\ell} = \begin{bmatrix} \dfrac{\partial \mathrm{L}_\ell}{\partial x} & \dfrac{\partial \mathrm{L}_\ell}{\partial y} & \dfrac{\partial \mathrm{L}_\ell}{\partial z} & \dfrac{\partial \mathrm{L}_\ell}{\partial \phi} & \dfrac{\partial \mathrm{L}_\ell}{\partial \theta} \end{bmatrix} \tag{47}
$$

and

$$
\delta\mathbf{q} = \begin{bmatrix} \delta x & \delta y & \delta z & \delta\phi & \delta\theta \end{bmatrix}^{\mathrm{T}} \tag{48}
$$

are the Jacobian matrix for the left actuator and the variation of generalized coordinates, respectively.

Replacing the time derivatives by variations in equation (44), the component of the Jacobian matrix can be obtained as

$$
\frac{\partial \mathrm{L}_\ell}{\partial x} = \frac{1}{\mathrm{L}_\ell}\left\{x + \left(1 - \frac{g}{\ell}\right)s(\theta)e - \frac{2g}{\ell}x + \frac{f}{\ell}c(\phi)c(\theta)e\right\} \tag{49}
$$

$$
\frac{\partial \mathrm{L}_\ell}{\partial y} = \frac{1}{\mathrm{L}_\ell}\{y + c(\phi)d - s(\phi)c(\theta)e - d\} \tag{50}
$$

$$
\begin{aligned}
\frac{\partial \mathrm{L}_\ell}{\partial z} &= \frac{1}{\mathrm{L}_\ell}\left\{\left(1 - \frac{2g}{\ell}\right)z + \left(1 - \frac{2g}{\ell}\right)s(\phi)d\right. \\
&\quad \left. + \left(1 - \frac{g}{\ell}\right)c(\phi)c(\theta)e + \frac{f}{\ell}s(\theta)e\right\}
\end{aligned}
\tag{51}
$$

$$
\begin{aligned}
\frac{\partial L_\ell}{\partial \phi} = \frac{1}{L_\ell} \Big\{ & [zc(\phi) - ys(\phi)]d - [zs(\phi) + yc(\phi)]c(\theta)e \\
& - \frac{2g}{\ell} c(\phi)d[z + ds(\phi)] + ec(\phi)d\left(\frac{f}{\ell}s(\theta) + c(\theta)\right) + d^2 s(\phi) \\
& + \frac{g}{\ell} ec(\theta)d[s^2(\phi) - c^2(\phi)] + e\left(\frac{f}{\ell}x + \frac{g}{\ell}z\right)s(\phi)c(\theta)\Big\}
\end{aligned} \tag{52}
$$

and

$$
\begin{aligned}
\frac{\partial L_\ell}{\partial \theta} = \frac{1}{L_\ell} \Big\{ & \left(1 - \frac{g}{\ell}\right)xc(\theta)e - \left(1 - \frac{g}{\ell}\right)zc(\phi)s(\theta)e + (y - d)s(\phi)s(\theta)e \\
& + e\left(\frac{f}{\ell}c(\theta) + \frac{g}{\ell}c(\phi)s(\theta)\right)s(\phi)d + \frac{f}{\ell}e[zc(\theta) + xc(\phi)s(\theta)]\Big\}
\end{aligned} \tag{53}
$$

Driving Constraint and Its Jacobian Matrix for the Right Actuator

The attachment point of the right actuator on the push beam is represented by a position vector $\mathbf{P}_9$ as

$$
\begin{aligned}
\mathbf{P}_9 \cong \mathbf{P}_5 + \frac{g}{\ell}\begin{bmatrix} x_6 \\ 0 \\ z_6 \end{bmatrix} + \frac{f}{\ell}\begin{bmatrix} -z_6 \\ 0 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 2y\frac{g}{\ell} \\ 0 \end{bmatrix} \\
= \begin{bmatrix} \frac{g}{\ell}x - \frac{f}{\ell}[z - s(\phi)d] \\ -d + 2y\frac{g}{\ell} \\ \frac{g}{\ell}[z - s(\phi)d] + \frac{f}{\ell}x \end{bmatrix}
\end{aligned} \tag{54}
$$

where the term $\begin{bmatrix} 0 & 2y\frac{g}{\ell} & 0 \end{bmatrix}^T$ is the approximated variation on the attachment point due to the roll motion of the blade.

The attachment point $\mathbf{P}_{10}$ on the blade, as shown in figure 1 is

$$
\mathbf{P}_{10} = \mathbf{r}_B + \mathbf{A}\mathbf{s}'_{10} \tag{55}
$$

where $\mathbf{s}'_{10}$ is a constant vector as

$$
\mathbf{s}'_{10} = \begin{bmatrix} 0 & -d & e \end{bmatrix}^T \tag{56}
$$

Following the same procedure described above for left actuator, the desired length and its velocity of the right actuator can be derived in the same way.

Since a perturbation method is used and evaluated at the nominal operating point where $\phi = 0$, the Jacobian matrix associated with the right tilt actuator can be obtained by simply replacing "d" in the equations (49-53) by "-d".

Driving Constraint and Its Jacobian Matrix for the Center Lift Actuator

The position vectors $\mathbf{P}_{12}$ and $\mathbf{P}_4$ on the chassis reference frame are

$$
\mathbf{P}_{12} = \begin{bmatrix} c1 & 0 & c2 \end{bmatrix}^T \tag{57}
$$

and

$$
\mathbf{P}_4 = \mathbf{r}_B = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{58}
$$

The position vector of the lift actuator can be computed as

$$
\mathbf{L}_c = \begin{bmatrix} x - c1 \\ y \\ z - c2 \end{bmatrix} \tag{59}
$$

and its square length is

$$
L_c^2 = (x - c1)^2 + y^2 + (z - c2)^2 \tag{60}
$$

Taking the time derivative of equation (60) yields

$$
\dot{L}_c = \begin{bmatrix} \frac{x - c1}{L_c} & \frac{y}{L_c} & \frac{z - c2}{L_c} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{61}
$$

or taking the variation of equation (60) to obtain

$$
\delta L_c = \begin{bmatrix} \frac{x - c1}{L_c} & \frac{y}{L_c} & \frac{z - c2}{L_c} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} \tag{62}
$$

Equation (62) can be written as

$$\delta L_c = \begin{bmatrix} \frac{x-c1}{L_c} & \frac{y}{L_c} & \frac{z-c2}{L_c} & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta\phi \\ \delta\theta \end{bmatrix} = \boldsymbol{\Phi}_{\mathbf{q}}^{L_c} \delta\mathbf{q} \tag{63}$$

The Jacobian matrix for the center lift actuator is therefore

$$\boldsymbol{\Phi}_{\mathbf{q}}^{L_c} = \begin{bmatrix} \frac{x-c1}{L_c} & \frac{y}{L_c} & \frac{z-c2}{L_c} & 0 & 0 \end{bmatrix} \tag{64}$$

Construction of Jacobian Matrix MO2I

The variation of actuator lengths and the variation of generalized coordinates can be constructed using equations (49-53) and (63) as

$$\delta\mathbf{L} = \begin{bmatrix} \delta L_\ell \\ \delta L_c \\ \delta L_r \end{bmatrix} = \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{MO2I}} \delta\mathbf{q} \tag{65}$$

where the Jacobian matrix MO2I (Matrix from Outer to Inner -- hydraulics actuators are in the inner loop while balde positions and velocities are in the outer loops of Control Block Diagram) that transfers the variations of generalized coordinates to the variations of actuators is

$$\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{MO2I}} = \begin{bmatrix} \boldsymbol{\Phi}_{\mathbf{q}}^{L_\ell} \\ \boldsymbol{\Phi}_{\mathbf{q}}^{L_c} \\ \boldsymbol{\Phi}_{\mathbf{q}}^{L_r} \end{bmatrix} = \begin{bmatrix} \frac{\partial L_\ell}{\partial x} & \frac{\partial L_\ell}{\partial y} & \frac{\partial L_\ell}{\partial z} & \frac{\partial L_\ell}{\partial \phi} & \frac{\partial L_\ell}{\partial \theta} \\ \frac{x-c1}{L_c} & \frac{y}{L_c} & \frac{z-c2}{L_c} & 0 & 0 \\ \frac{\partial L_r}{\partial x} & \frac{\partial L_r}{\partial y} & \frac{\partial L_r}{\partial z} & \frac{\partial L_r}{\partial \phi} & \frac{\partial L_r}{\partial \theta} \end{bmatrix}_{3\mathrm{x}5} \tag{66}$$

The relationship between the velocities of actuators and blade can be obtained by simply replacing the variations with the time derivatives as

$$\dot{\mathbf{L}} = \begin{bmatrix} \dot{L}_\ell \\ \dot{L}_c \\ \dot{L}_r \end{bmatrix} = \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{MO2I}} \dot{\mathbf{q}} \tag{67}$$

The velocities of the actuators are used to compute desired valve commands in the control algorithm.

Construction of Jacobian Matrix MI2O

The displacements and velocities of actuators depend on the performance of the hydraulic system and the control algorithm and may not match the desired values commanded. Therefore, the resulting motion of blade must be evaluated based on the actual movement of the actuators. In other words, the Jacobian matrix MI2O that transfers the variations of actuator lengths to the variations of generalized coordinates should be constructed.

It is clear that three variations of actuator lengths can't determine the five variations of generalized coordinates, because the actuators control only three degrees of freedom of the blade, while the remaining two are decided by the two joint constraints no matter how the hydraulic system performs. Therefore, the joint constraints in equations (27-28) must be satisfied and their Jacobian must be taken into account.

Combining the Jacobian matrices for the actuators shown in equation (65) and for the joint constraints shown in equation (29) yields

$$\begin{bmatrix} \delta L_\ell \\ \delta L_c \\ \delta L_r \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{MO2I}}{}_{3\mathrm{x}5} \\ \boldsymbol{\Phi}_{\mathbf{q}_{2\mathrm{x}5}}^{J} \end{bmatrix} \delta\mathbf{q}$$

$$= \begin{bmatrix} \frac{\partial L_\ell}{\partial x} & \frac{\partial L_\ell}{\partial y} & \frac{\partial L_\ell}{\partial z} & \frac{\partial L_\ell}{\partial \phi} & \frac{\partial L_\ell}{\partial \theta} \\ \frac{x-c1}{L_c} & \frac{y}{L_c} & \frac{z-c2}{L_c} & 0 & 0 \\ \frac{\partial L_r}{\partial x} & \frac{\partial L_r}{\partial y} & \frac{\partial L_r}{\partial z} & \frac{\partial L_r}{\partial \phi} & \frac{\partial L_r}{\partial \theta} \\ 0 & 1 & 0 & -ds(\phi) & 0 \\ x & y & z & d^2 s(\phi) & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta\phi \\ \delta\theta \end{bmatrix} \tag{68}$$

The perturbations $\frac{\partial L_\ell}{\partial \theta}$ or $\frac{\partial L_r}{\partial \theta}$ is observed from equation (53) to be non zero for operation range $\theta < 90^0$, and the first and third row in equation (68) are linear independent. Therefore, the coefficient matrix is nonsingular and its inverse exists. Equation (68) can be rewritten as

$$\begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta\phi \\ \delta\theta \end{bmatrix} = \begin{bmatrix} \frac{\partial L_\ell}{\partial x} & \frac{\partial L_\ell}{\partial y} & \frac{\partial L_\ell}{\partial z} & \frac{\partial L_\ell}{\partial \phi} & \frac{\partial L_\ell}{\partial \theta} \\ \frac{x-c1}{L_c} & \frac{y}{L_c} & \frac{z-c2}{L_c} & 0 & 0 \\ \frac{\partial L_r}{\partial x} & \frac{\partial L_r}{\partial y} & \frac{\partial L_r}{\partial z} & \frac{\partial L_r}{\partial \phi} & \frac{\partial L_r}{\partial \theta} \\ 0 & 1 & 0 & -ds(\phi) & 0 \\ x & y & z & d^2 s(\phi) & 0 \end{bmatrix}^{-1} \begin{bmatrix} \delta L_\ell \\ \delta L_c \\ \delta L_r \\ 0 \\ 0 \end{bmatrix} \tag{69}$$

or in compact form

$$\delta \mathbf{q} = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta\phi \\ \delta\theta \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}}^{MO2I}{}_{3x5} \\ \mathbf{\Phi}_{\mathbf{q}}^{J}{}_{2x5} \end{bmatrix}^{-1} \begin{bmatrix} \delta L_\ell \\ \delta L_c \\ \delta L_r \\ 0 \\ 0 \end{bmatrix} = \mathbf{\Phi}_{\mathbf{q}}^{MI2O} \begin{bmatrix} \delta L_\ell \\ \delta L_c \\ \delta L_r \\ 0 \\ 0 \end{bmatrix} \tag{70}$$

where

$$\mathbf{\Phi}_{\mathbf{q}}^{MI2O}{}_{5x5} = \begin{bmatrix} \mathbf{\Phi}_{\mathbf{q}}^{MO2I}{}_{3x5} \\ \mathbf{\Phi}_{\mathbf{q}}^{J}{}_{2x5} \end{bmatrix}^{-1} \tag{71}$$

is the Jacobian matrix MI2O.

The relationship between the velocities of blade and actuators can be obtained by simply replacing the variations in equation (70) with the time derivatives as

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{L}_\ell \\ \dot{L}_c \\ \dot{L}_r \end{bmatrix} = \mathbf{\Phi}_{\mathbf{q}}^{MI2O} \dot{\mathbf{L}} \tag{72}$$

As described in the section GOAL AND ASSUMPTIONS, the goal of this derivation is to determine velocity relationship between the actuator and blade movements in order to issue valve commands for hydraulic system. The controller design, hydraulic flow distribution among actuators, or insufficient flow issues are beyond the scope of this paper. Assume the desired blade speed $\dot{\mathbf{q}}$ is determined in the control loop, equations (67) computes the required actuator velocities $\dot{\mathbf{L}}$. These velocities determine flow requirements and distributions and thus valve opening commands. The Jacobian matrix $\mathbf{\Phi}_{\mathbf{q}}^{MO2I}$ that transforms outer loop quantities $\dot{\mathbf{q}}$ to inner loop quantities $\dot{\mathbf{L}}$ is a function of position. Depending on computational speed of microprocessor, it might be updated each sample time step, or held constant near nominal operation point.

On the other hand, equation (72) predicts the blade moving velocities based on the commanded actuator velocities. In the early stage of control algorithm development, it can be used to update the generalized coordinates of blade to predict the next time step position inside PLANT. In the middle or final development stage, it may be replaced by

(1) a multi-body dynamics software (e.g., DADS) that can be integrated with control/hydraulics for off-line simulation. In this case, the velocities and lengths of actuators can be directly feedback from the dynamics output.

(2) linear transducers for actuators in Real-Time operation that provide lengths and derivatives (velocities) for control feedbacks.

VALIDATION AND IMPLEMENTATION

The formulation developed above has been implemented in the prototype vehicles and validated by comparing the results with field measured data as follows:

The field measured data for the chassis frame, blade frame, and center tine tip position of the Combat Vehicle is shown in figure 4.

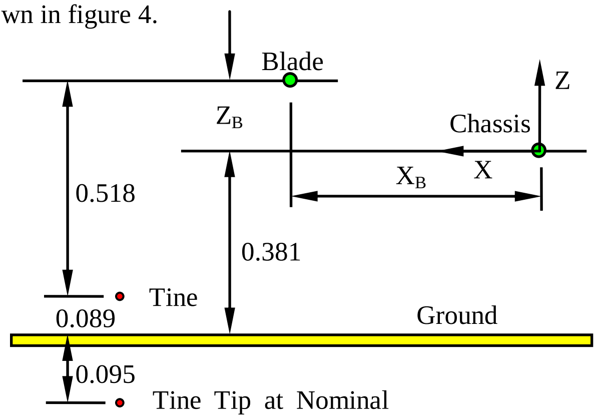

Figure 4: Measured Data for MCB

when the tine tip is above ground by 0.089 m in the configuration of figure 4,

$$Z_B = (0.518 + 0.089) - 0.381 = 0.226 \text{ (m)}$$

$$X_B = \sqrt{(1.26)^2 - (0.226)^2} = 1.23957 \text{ (m)}$$

with roll and pitch angles near zero, the lengths of three actuators are

$L_R = 1.0183\text{m}$, compared with 1.016 m from measured data.

$L_C = 0.9783\text{m}$, compared with 0.981 m from measured data.

$L_L = 1.0183\text{m}$, compared with 1.016 m from measured data.

Figure 5 shows the length of center lift actuator as function of blade height. The operation range is within the maximum and minimum actuator stroke.

Figure 6 shows the length of left or right tilt actuator as function of blade height. The operation range is within the maximum and minimum actuator stroke.

Figure 7 shows the perturbation of Blade Z vs. lift actuator stroke at different blade heights. For example, the perturbation ratio is -3.75 at blade height -0.1 meter means that one unit lift actuator extension will result in 3.75 units blade vertical downward displacement at the specified configuration.

Figure 8 shows the perturbation of tilt vs. lift actuator stroke at different blade height. For instance, the perturbation ratio is -1.49 at blade height -0.1 meter means that one unit lift actuator extension requires 1.49 units tilt actuator retract in order to maintain the pitch angle of blade equals to zero at the specified configuration. This data is used in the flow distribution in the hydraulic system to control the blade angle.

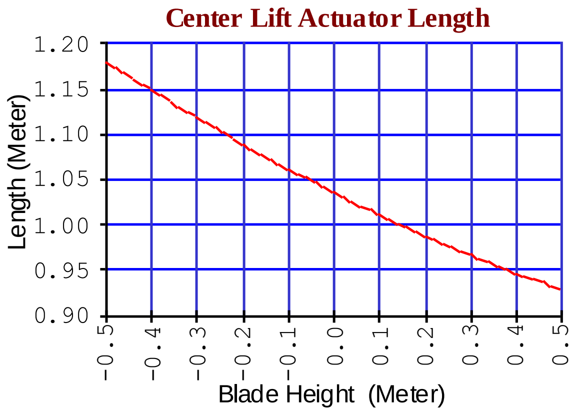

Figure 5: Center Lift Actuator Length as Function of Blade Height

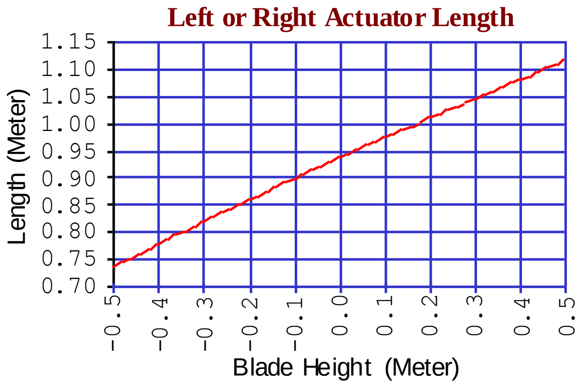

Figure 6: Left or Right Tilt Actuator Length as Function of Blade Height

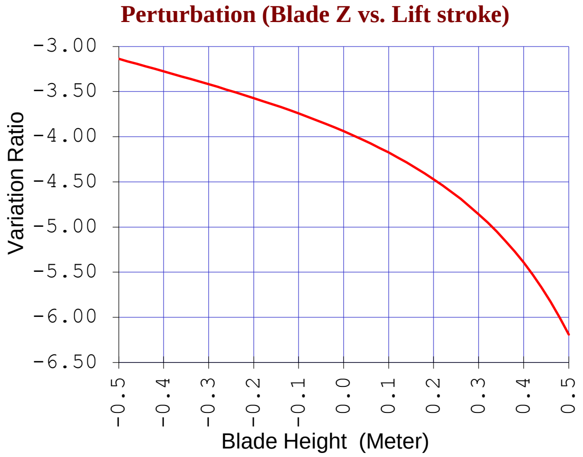

Figure 7: Perturbation of Blade Z vs. Lift Actuator Stroke as Function of Blade Height

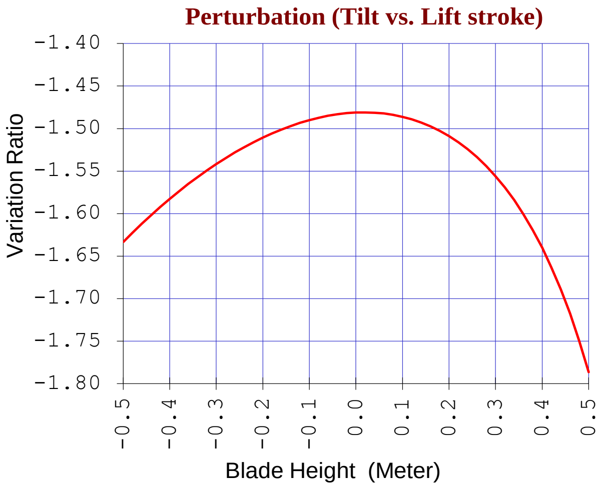

Figure 8: Perturbation of Tilt vs. Lift Actuator Stroke as Function of Blade Height

CONCLUSION

A kinematic analysis for automatic plowing depth control of a Mine Clearing Combat Vehicle has been developed and the formulation has been implemented in two prototypes. The computational result is used in the control algorithm to issue valve commands for flow distribution in the hydraulic systems and has been validated by comparing with field measured data.

REFERENCES

[1] "DADS User's Manual," Computer Aided Design Software, Inc., Oakdale, Iowa. Or http://www.lmsintl.com/simulation/virtuallab/motion.

[2] ADAMS: http://www.mscsoftware.com/Submitted-Content/Resources/DS_MSCAdams_A4_w.pdf.

[3] Haug, E.J., 1989, "Computer Aided Kinematics and Dynamics, Vol I," Allyn & Bacon, Boston.