

# Planning Trajectories on Uneven Terrain using Optimization and Non-Linear time Scaling Techniques

Arun Kumar Singh, K Madhava Krishna and Srikanth Saripalli

**Abstract**—In this paper we introduce a novel framework of generating trajectories which explicitly satisfies the stability constraints such as no-slip and permanent ground contact on uneven terrain. The main contributions of this paper are: (1). It derives analytical functions depicting the evolution of the vehicle on uneven terrain. These functional descriptions enable us to have a fast evaluation of possible vehicle stability along various directions on the terrain and this information is used to control the shape of the trajectory. (2) It introduces a novel paradigm wherein non-linear time scaling brought about by parametrized exponential functions are used to modify the velocity and acceleration profile of the vehicle so that these satisfy the no-slip and contact constraints. We show that non-linear time scaling manipulates velocity and acceleration profile in a versatile manner and consequently has exceptional utility not only in uneven terrain navigation but also in general in any problem where it is required to change the velocity of the robot while keeping the path unchanged like collision avoidance.

## I. INTRODUCTION

Trajectory generation for mobile robot on uneven terrain is a challenging problem because the notion of the vehicle stability governed by the vehicle-terrain interaction has to be included in the planning framework. The primary requirement for calculating stability on uneven terrain is to calculate the position and attitude of the vehicle as it evolves on the terrain. However for a passive suspension car-like vehicle, not all states can be directly controlled. The proposed work gives closed form functional relations which gives the evolution of the vehicle's passive state in terms of actively controlled states. With this important feature in place, it becomes possible to quickly analyse the velocity and acceleration profiles of the trajectories and check whether they satisfy the stability constraints. This forms the crux of the proposed trajectory generation framework which follows a two-step process.

At the first step the vehicle trajectory between a start and a goal location is generated in the form of parametric functions, without considering the vehicle dynamics. At the second step the vehicle dynamics is considered and a non-linear scaling transformation is proposed for trajectories such that those satisfy the stability constraints. A single parameter called  $s_{min}$  is proposed whose existence decides whether a given trajectory can be modified by non-linear scaling transformation to satisfy the stability constraints.

K Madhava Krishna is with Robotics Research Centre, IIIT-Hyderabad, India [mkrishna@iiit.ac.in](mailto:mkrishna@iiit.ac.in)

Arun Kumar Singh is PhD candidate at Robotics Research Lab, IIIT-Hyderabad [arunkumar.singh@research.iiit.ac.in](mailto:arunkumar.singh@research.iiit.ac.in)

Srikanth Saripalli is with ASTRIL, Arizona State University [srikanth.saripalli@asu.edu](mailto:srikanth.saripalli@asu.edu)

For Trajectories which cannot be modified to satisfy the stability constraints, a trajectory replanning framework is embedded in the planner. The replanning framework builds on the fact that the parametric trajectory generation framework proposed in this paper generates multiple trajectories between a given start and goal location. Moreover as shown in section III the shape of the trajectories can be controlled to pass through stable regions.

## II. RELATED WORK

The problem of generating stable trajectories on uneven terrain has been addressed in the past by many researchers. For example in [3,4] the planner produces trajectories considering postural stability. However we have shown in [5] that postural stability measured in terms of variation of pitch and roll angle proves to be inadequate for motion planning purposes. For this reason a new metric was proposed in [5] which measures vehicle stability by noting how well the current velocity and acceleration profile satisfy the no-slip and permanent contact constraint. The idea of using no-slip and permanent contact constraint in the motion planning algorithm has been exploited in the past in [6,7]. However the key novelty of the current proposed work is that it provides vehicle dynamics in 3D. This is in contrast to the point mass [6] and planar model of [7].

The non-linear scaling transformation mentioned in the previous section is an improvement over the constant scaling transformation proposed in [1]. Scaling transformation is an elegant mathematical technique which brings about a transformation in the independent variable of the function affecting its derivative while keeping the path taken by the function unchanged. Due to this mathematical elegance, it finds its way naturally in controlling the vehicle dynamics along a given path. Hence in [1], it was exploited to make the velocity and acceleration profile of the manipulator trajectories satisfy the actuator bounds. In [2] the scaling concept was used to generate stable trajectories for mobile robots on planar uneven terrains. Both [1,2] however use a very simple scaling concept which can be represented in the following form:

$$\frac{du}{dt} = s \Rightarrow u = st \quad (1)$$

(1) gives a transformation from the variable  $u$  to  $t$ . It can be seen that  $t$  is just a scalar multiple of  $u$  and hence derivative of any function  $g(u)$  gets scaled by a constant factor  $s$ . However there is a critical problem with such transformations which severely limits its use. To understand this further, suppose that some function  $g(u)$  shown in

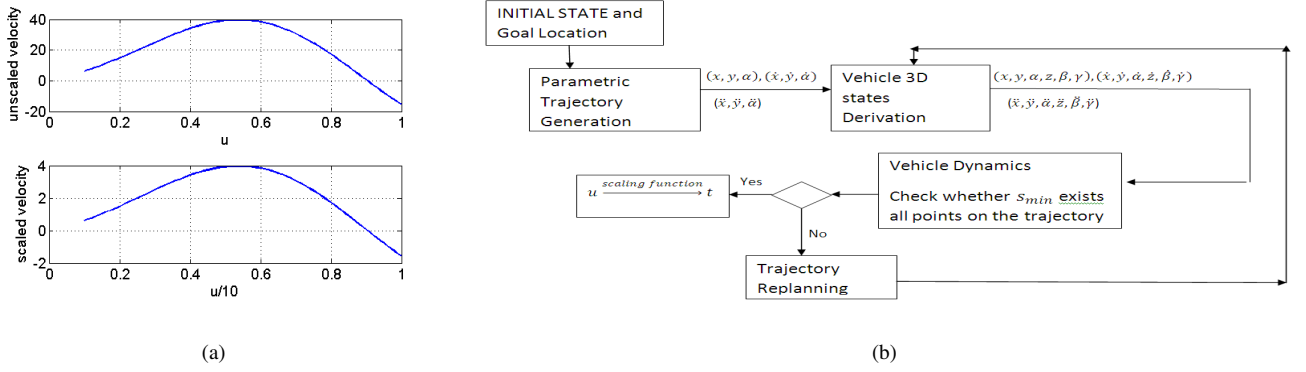


Fig. 1. (a):Problem with constant scaling. (b) System overview

figure.1(a) describes the velocity of a vehicle and we employ the above scaling transformation to modify it. It can be seen that the scaled and unscaled velocity profile does not start at the same point which means that the a constant scaling will bring a discontinuity in the velocity space at the initial point. Another interpretation of this can be that it is only possible to use constant scaling transformation at a planning level and not in a reactive manner while executing a trajectory. For example midway while executing a trajectory, if it is required to change for any reason the velocity and acceleration profile while keeping the path unchanged, constant scaling [1] cannot be used. Non-linear scaling transformation on the other hand can adequately handle situations like these. Obviously the problem of velocity discontinuity does not arise when the initial unscaled velocity profile starts from zero which is exactly what is being used in [1,2]. However on a 3D uneven terrain a trajectory generation framework which requires the vehicle to first de-accelerate to zero velocity is not appropriate.

To get rid of the above problems we propose a scaling transformation of the following form

$$\frac{du}{dt} = k_1 e^{-k_2 u} \quad (2)$$

We provide in later sections analytical expressions to determine constants  $k_1$  and  $k_2$  which controls the manipulation of velocity and acceleration profile in a far more flexible manner as compared to constant scaling described in (1).

Figure.1(b) gives an overview of the framework proposed in the current work. The parametric trajectory generation module shown in the figure is explained in section III. The derivation of vehicle 3D states and dynamics is provided in section IV. Non-linear time scaling is explained in section V along with trajectory replanning. Simulation results are presented in section VI.

### III. PARAMETRIC TRAJECTORY GENERATION

For a passive suspension car-like vehicle, only the yaw plane components i.e position  $x, y$  and heading  $\alpha$  can be controlled. All the other vehicle states are a function of yaw plane components and in later sections we derive the exact functional relationships between them. In this section we

derive the yaw plane trajectories as parametric functions like [8,9,10] for a given a start and goal location. Let us begin with parametrizing the derivatives of the states as

$$\dot{x}(u) = \sum_{i=1}^2 a_i \cos(i\omega u) + b_i \sin(i\omega u) \quad (3)$$

$$\zeta(u) = \tan \alpha = \sum_{i=1}^2 c_i \cos(i\omega u) + d_i \sin(i\omega u) \Rightarrow \dot{\alpha} = \frac{\dot{\zeta}}{1 + \zeta^2} \quad (4)$$

By the non-holonomic constraint of the robot we have

$$\dot{y} \cos \alpha - \dot{x} \sin \alpha = 0 \Rightarrow \dot{y} = \dot{x} \tan \alpha = \dot{x} \zeta \quad (5)$$

Integrating (3) and (5) we get the parametrised trajectory as  $\chi(\mathbf{u}) = (\mathbf{x}(\mathbf{u}), \mathbf{y}(\mathbf{u}))$ . The parameters  $a_i, b_i, c_i, d_i$  can be obtained by solving the following two point boundary value problem

$$\begin{cases} x(u_0) = x_0, y(u_0) = y_0, \zeta(u_0) = \zeta_0 \\ \frac{dx}{du}(u_0) = v_i, \frac{d\alpha}{du}(u_0) = \alpha_i \\ x(u_f) = x_f, y(u_f) = y_f, \zeta(u_f) = \zeta_f \end{cases} \quad (6)$$

(6) can be solved by putting it as an equality constraint in an optimization routine with the objective function as

$$L(x(u), y(u)) = \int_{u_0}^{u_f} \sqrt{\dot{x}^2 + \dot{y}^2} du \quad (7)$$

Figure 2 shows a sample output of optimisation routine. The various trajectories between the start and goal point are generated by varying the value of  $\zeta_f$  in (6). An important thing to note about figure 2 is that there are two sets of trajectories  $C^+$  and  $C^-$  approaching the goal from two different sides. The  $C^+$  trajectories corresponds to  $\zeta > 0$  and  $C^-$  corresponds to  $\zeta < 0$ . Another important aspect is that the shape of the trajectory can be controlled by the magnitude of  $\zeta$  with radius of curvature increasing with increasing  $|\zeta|$ . This feature is utilised to make the trajectory bend towards stable regions of the terrain. We have observed that defining the states as in (3)-(5) has improved the convergence of the optimization problem for large  $\zeta_f$  as compared to the performance of similar parametric function definitions found in [19]. Since range of tangent function is infinite,  $\zeta_f$  can attain any real value.

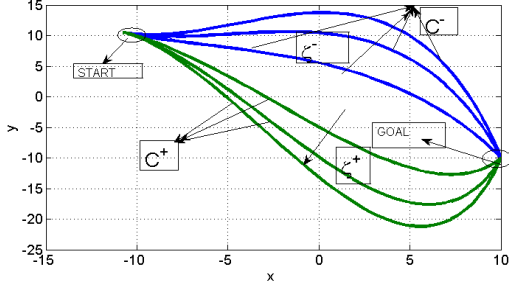


Fig. 2. Figure how paths with varying radius of curvatures can be generated by changing  $\zeta$ . Two distinct sets of shapes can be seen corresponding to  $\zeta > 0$  and  $\zeta < 0$

#### IV. DERIVATION OF VEHICLE 3D STATES AND DYNAMICS

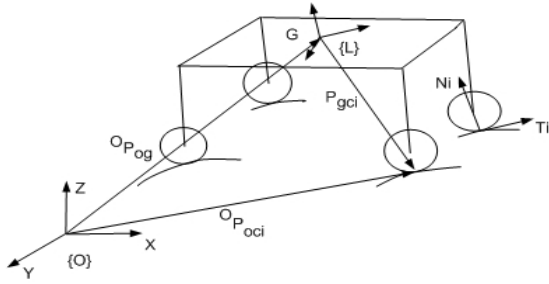


Fig. 3. A Car-like Vehicle

From the yaw plane trajectory just obtained in the previous section, we obtain the 3D trajectory of the vehicle as it evolves on an uneven terrain. We assume that the terrain equation can be known in the following form

$$a = f(b, c) \quad (8)$$

Where 'a' represents the terrain height at the  $x$  and  $y$  coordinate ( $a, b$ ). The assumption of known terrain equation in this form has been used in the past by many researchers [6,11,12,13] primarily because of the fact that such terrain equation can be generated by Data Elevation Models (DEM). Some significant paper addressing this issue can be found in [14,15,16]. With the help of terrain equation this section derives analytical functions relating  $x, y, \alpha$  to vehicle's  $z$  coordinate, roll  $\beta$  and pitch  $\gamma$ . To this effect consider the holonomic constraints defining the geometry of the vehicle (refer figure.3)

$$\vec{P}_{OG} + \vec{P}_{Gci} = \vec{P}_{Oci} \quad (9)$$

where  $\vec{P}_{Gci} = \vec{r}_f$

$$\vec{r}_f = R \begin{bmatrix} \delta h & \frac{2.5-i}{|(2.5-i)|} w & -(l_i + r) \end{bmatrix}, \forall i = 1, 2, 3, 4$$

$$\delta = 1, i = 1, 4$$

$$= -1, i = 2, 3$$

$R$  is the rotation matrix describing the orientation of the body fixed  $\{G\}$  with respect to frame moving with the body  $\{L\}$ .  $\{G\}$  has the same orientation as the inertial frame  $\{0\}$ .  $(2.5 - i)$  and  $\delta$  has been incorporated to ensure proper sign of  $w$  and  $h$  corresponding to each vertex of the chassis.  $l_i$  are the leg lengths.  $h$  and  $w$  are half width and breadth of the chassis and  $r$  is the radius of the wheels. (9) written for all

the wheels represents 12 equations in 15 variables. They are twelve wheel ground contact points  $x_{ci}, y_{ci}, z_{ci}$ , roll  $\beta$ , pitch  $\gamma$  and  $z$  coordinate of the vehicle. It is observed that for a passive/rigid suspension car-like vehicle, tip over instability is initiated for roll and pitch angle beyond 0.5 radian and the vehicle cannot remain in contact with the ground. In other words the variation of the roll and pitch angle along a stable path will be less than this limit and hence with this fact in mind, we linearise (9) with respect to  $\beta$  and  $\gamma$  for each wheel as:

$$x_{ci} = x - \frac{2.5 - i}{|(2.5 - i)|} w \sin \alpha - l_i \gamma \sin \alpha - \delta h \cos \alpha - l_i \beta \cos \alpha \quad (10)$$

$$y_{ci} = y + \frac{2.5 - i}{|(2.5 - i)|} w \cos \alpha - l_i \beta \sin \alpha - \delta h \sin \alpha + l_i \gamma \cos \alpha \quad (11)$$

$$z_{ci} = z + \frac{2.5 - i}{|(2.5 - i)|} w \gamma - l_i + \delta h \beta \quad (12)$$

$x_{ci}, y_{ci}, z_{ci}$  satisfy (8) and to explicitly solve for  $\beta$  and  $\gamma$  as a function of  $x, y, \alpha$  it is required that (8) could be represented as a combination of piecewise linear hyperplanes. In case when the fitted terrain equation to the DEM is non-linear we can linearise the terrain equation about the vehicle centre of mass. This linearisation is justified since any terrain can be locally represented by a linear plane having a particular orientation in 3D space. Linearising (8) about the current chassis centre coordinate gives

$$z_{ci} = k_3 + k_1(x_{ci} - x) + k_2(y_{ci} - y) \quad (13)$$

where  $k_3 = f(x, y)$ ,  $k_1 = \frac{\partial(f)}{\partial b}$ ,  $b = x$ ,  $c = y$ ,  $k_2 = \frac{\partial(f)}{\partial c}$ ,  $b = x$ ,  $c = y$ .

Substituting  $x_{ci}, y_{ci}, z_{ci}$  values from (10), (11), and (12), four equations represented by (13) can be written in the matrix form as

$$\begin{bmatrix} 1 & w + p_1 & h + q_1 \\ 1 & w + p_2 & -h + q_2 \\ 1 & -w + p_3 & -h + q_3 \\ 1 & -w + p_4 & h + q_4 \end{bmatrix} \begin{bmatrix} z \\ \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{bmatrix} \quad (14)$$

$p_i, q_i, H_i$  are given in the appendix. The coefficient matrix in (14) can be pseudo-inverted to solve for  $z, \beta, \gamma$ . However if the suspension travel length is small which essentially means that  $p_1 = p_2 = p_3 = p_4 = p$ , and  $q_1 = q_2 = q_3 = q_4 = q$ , with small matrix manipulation (14) can be reduced to

$$\begin{bmatrix} 0 & 2w & 0 \\ 0 & 0 & -2h \\ 1 & -w + p & h + q \end{bmatrix} \begin{bmatrix} z \\ \gamma \\ \beta \end{bmatrix} = \begin{bmatrix} H_2 - H_3 \\ H_3 - H_4 \\ H_4 \end{bmatrix} \quad (15)$$

Inverting coefficient matrix in (15) gives  $z, \gamma, \beta$  as analytical functions of  $x, y, \alpha$ . Figure 4 shows that linearised posture determination model agrees well with that obtained from solving (9) in the non-linear form. These relations and their derivatives are imperative for obtaining 3D dynamics of the robot and estimating stable velocities and accelerations.

(10)-(12) will allow us to have wheel ground contact points also as a function of  $x, y$  and  $\alpha$ . Wheel ground contact points location are important because they decide contact surface unit normals  $n_{xi}, n_{yi}, n_{zi}$  and with the help of that the

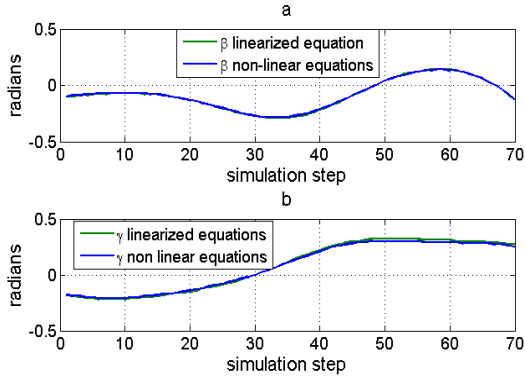


Fig. 4. Comparison of posture parameters obtained by (15) with that obtained by solving (9) in non-linear form along an arbitrary path on a 3D terrain. Good agreement between the linear and non-linear model justifies our linearization

traction unit vectors can be calculated. The details of these derivations can be found in [5]. The equations of motion of the vehicle can be presented in the form similar to [5,17].

$$A * C = D \quad (16)$$

where  $C = [T_i \ N_i]_{2n \times 1}^T$ ,  $D = [\vec{a} \ \vec{\Omega}]_{6 \times 1}^T$ .  $T_i, N_i, \vec{a}, \vec{\Omega}$  are the traction, normal forces, linear and angular accelerations respectively.  $n$  is the number of wheels of the vehicle. The elements of  $A_{6 \times 2n}$  matrix depends on vehicle posture, geometry and terrain dependent parameters like surface contact normals and traction unit vectors. It is to be noted that because of the derivation presented from (10-15) matrix  $A$  can also be known in closed form as a function of  $x, y, \alpha$ . Ideally if matrix  $A$  could be inverted symbolically, we could have analytical functional description of the variation of  $T_i, N_i$  with respect to  $\vec{a}, \vec{\Omega}$  and in theory we could have gradient descent based algorithm to generate a one shot stable path. Vehicles operating on uneven terrain generally have 4-8 wheels which makes  $A$  under-constrained and have to be pseudo-inverted. Some algorithms like [18] computes symbolic pseudo inverse for small matrices having one or two independent variables. However pseudo-inverting matrix  $A$ , whose dimension will increase with the number of wheels can turn into a complex problem. It should be noted that this problem is not unique to the framework proposed in this paper but is fundamental with modelling vehicle dynamics in 3D and relating traction and normal forces to velocity and acceleration in closed form. Hindered by this critical problem, a two step process of trajectory generation stated earlier in section I is followed which utilises the fact that it is relatively easy to compute the pseudo inverse numerically at any point on the terrain and by doing so we get

$$\begin{aligned} T_i = & a_{i1}(ma_x) + a_{i2}(ma_y) \\ & + a_{i3}(mg + ma_z) + a_{i4}(I_{xx}\dot{\Omega}_x) \\ & + a_{i5}(I_{yy}\dot{\Omega}_y) + a_{i6}(I_{zz}\dot{\Omega}_z) \end{aligned} \quad (17)$$

$$\begin{aligned} N_i = & a_{j1}(ma_x) + a_{j2}(ma_y) \\ & + a_{j3}(mg + ma_z) + a_{j4}(I_{xx}\dot{\Omega}_x) \\ & + a_{j5}(I_{yy}\dot{\Omega}_y) + a_{j6}(I_{zz}\dot{\Omega}_z) \end{aligned} \quad (18)$$

$\forall i = \{1, 2, 3, 4\}, \forall j = \{5, 6, 7, 8\}$ .  $m$  is the mass of the vehicle and  $g$  is acceleration due to gravity.  $a_{i1}, a_{i3}, a_{j2}, \dots, a_{in}, a_{jn}$ , are the elements of the pseudo inverse matrix of  $A$ . A feasible set of linear and angular accelerations is determined by the elements of the pseudo-inverse matrix and is defined as one which satisfies the following constraints

$$N_i > 0 \quad (19)$$

$$|(T_i)| < \rho N_i \quad (20)$$

$\rho$  is the coefficient of friction in (20). By noting (17) and (18) it can be seen that configuration  $x, y, \alpha$  which lead to  $a_{j3} > 0$  are promising candidate for satisfying (19) and (20) (since this coefficient decides the contribution of the weight). These coefficients can be rapidly evaluated for some look ahead distance along various directions and this information is fed to the trajectory generation described in the previous section to control the shape of the trajectory between start and a goal location.

Once a parametric trajectory has been obtained it is fed to (17) and (18) and check for the satisfaction of (19) and (20). Let  $s_{min}$  denote the minimum magnitude by which the velocity and acceleration profile of the trajectory needs to be scaled so that they satisfy (19) and (20). Unfortunately not all trajectories can be scaled to satisfy (19) and (20) and this fact gives rise to a novel definition of stability.

*Definition:* A Trajectory can be modified to satisfy the (19) and (20) if at all points along the trajectory a  $s_{min}$  could be found. Those trajectories will be referred to as stable.

Trajectories for which  $s_{min}$  could be found at every point, a transformation is brought in the parametric definition such that the velocity and acceleration profile is scaled by the amount suggested by  $s_{min}$ . This is explained in detail in the next section.

## V. NON-LINEAR TIME SCALING

A transformation from  $u$  to  $t$  in the independent variable of the function, following the definition of (2) brings about the following change in the derivative of the states.

$$\dot{x}(t) = \dot{x}(u) \frac{du}{dt}, \ddot{x}(t) = \left(\frac{du}{dt}\right)^2 \ddot{x}(u) + \dot{x}(u) \frac{d^2u}{dt^2} \quad (21)$$

The changes in the derivatives of  $y, \alpha$  follow the same form as (21). If  $\frac{du}{dt} < 1$ , the velocity and acceleration profile is scaled down and vice versa for  $> 1$ . The scaling function  $\frac{du}{dt}$  should be such that the following inequality is satisfied.

$$\begin{aligned} \frac{du}{dt} & \leq s_{min}(u), \text{ if } s_{min}(u) \leq 1 \\ \frac{du}{dt} & \geq s_{min}(u), \text{ if } s_{min}(u) \geq 1 \end{aligned} \quad (22)$$

(23) should be read with respect to scaling the velocity and acceleration profile up or down. In case of scaling down, the scaling function should be such that it scales down the profile by magnitude atleast equal to  $s_{min}$ . Similarly for scaling up.

To construct the scaling function  $\frac{du}{dt}$ , the key ingredients are the constants  $k_1$  and  $k_2$ . To have an insight into the construction of the scaling function, consider a simple case where it is required to scaled the velocity profile at two places i.e let  $s_{min} = s_1, u = u_a$  and  $s_{min} = s_2, u = u_b$ . If the constants  $k_1$  and  $k_2$  are chosen as below, we have a scaling function which scales the velocity and acceleration profile at the mentioned two points.

$$k_2 = \frac{\ln(\frac{s_1}{s_2})}{u_b - u_a}, k_1 = e^{k_2 u_a} \quad (23)$$

To build a more complicated scaling function which scales velocity and acceleration profile at any number of required points, a combination of continuously connected functions of the same form as (2) is chosen with the constants  $k_1, k_2$  determined in the same way as explained above.

#### A. Trajectory Replanning

As stated earlier, if for some portions of the initial planned trajectory, no  $s_{min}$  which would lead to satisfaction of (19) and (20), then the path is considered to be unstable. In such cases it is required that the planner quickly re-plans the infeasible portion of the trajectory. Let  $u_r$  correspond to some point on the trajectory before the point till which we can find a scaling factor. From that point onwards the optimization problem represented by (6-7) is solved in the domain  $u_r \leq u \leq u_f$  ensuring the continuity in the velocity and position space. Multiple paths are generated similar to figure. 2. Here also the possible vehicle stability along various directions as measured by the variation of the coefficient  $a_{j3}$  described in section IV is used as a guiding factor to control the shape of the trajectory. We generate multiple paths passing through the regions having appropriate  $a_{j3}$ . The final feasible trajectory will be a concatenation of many continuously connected stable trajectories.

### VI. SIMULATION RESULTS

The framework described above has been used here to generated trajectories on 3D uneven terrain. A small vehicle model with  $m = 3kg, \rho = 0.7$  and with dimension  $1 \times 1 \times 1m^3$  was used in the simulation. The value of  $\omega$  in parametric trajectory definition was chosen as  $\pi/10$ . The terrain is modelled as  $z = 3.5(0.4\cos(0.3x) + 0.5\sin(0.2y))$

#### A. Example

At the first step the coefficient  $a_{j3}$  were evaluated along various directions which provided good initial guess for stable regions of the terrain. Multiple trajectories were generated through the stable regions as can be seen from figure. 5(a). Then along the points of the obtained trajectories existence of  $s_{min}$  was checked. Figure.5(b) shows the  $s_{min}$  profile along the trajectories. As can be seen that for trajectories 1,2  $s_{min}$  does not exist at all points and hence by definition these are un-stable. For trajectory 4 on the other

hand  $s_{min}$  exists at all points and hence is stable. Figure.5(c) shows the evolution of the vehicle on the uneven terrain. Corresponding to the  $s_{min}$  profile obtained for trajectory 4. two scaling functions shown in figure.5(d) were created. The two scaling functions are defined in the following manner. Scaling function 1

$$\frac{du}{dt} = \begin{cases} e^{-1.42u}, & 0 \leq u \leq 2 \\ \frac{1}{17}, & 2 \leq u \leq 10 \end{cases} \quad (24)$$

Scaling function 2

$$\frac{du}{dt} = \begin{cases} e^{-1.42u}, & 0 \leq u \leq 2 \\ \frac{1}{17}, & 2 \leq u \leq 3.1 \\ 0.0011e^{1.27u}, & 3.1 \leq u \leq 4.6 \\ 31.61e^{-0.95u}, & 4.6 \leq u \leq 6.6 \\ \frac{1}{17}, & 6.6 \leq u \leq 8.1 \\ 6.4110^{-6}e^{1.12u}, & 8.1 \leq u \leq 10 \end{cases} \quad (25)$$

To understand how these scaling functions were created note from figure.5(d) that  $s_{min} = 1, u = 0, s_{min} = 0.059, u = 2$ . To satisfy these requirements and also (22), we use equation (23) to calculate  $k_1$  and  $k_2$  as 1 and -1.42 respectively. From that point onwards a constant scaling function can be used i.e  $\frac{du}{dt} = 0.059$  satisfies (22) for the concerned  $s_{min}$  profile. Figure.5(f) shows the velocity profile with scaling function 1. By comparing it with the unscaled velocity profile shown in figure.5(e), it can be seen that the time required to execute the trajectory has increased from 10s to 147.3s ( $u = (0, 10) \rightarrow t = (0, 147.3)$ ).

A more optimum velocity profile can be obtained if multiple exponential functions are used as in scaling function 2. As can be seen from figure.5(d), at intervals where  $s_{min} = 1(3.1 \leq u \leq 6.6, 8.1 \leq u \leq 10)$ , the scaling function is allowed to increase. As a direct consequence of the magnitude of the velocity profile becomes higher as can be seen from figure.5(g). It is to be noted that the time required to execute the trajectory has decreased from 147.3s to 96.3s. Thus non-linear scaling provides a greater level of flexibility in modifying the velocity and acceleration profile as compared to the constant scaling [1]. It is also to be noted that the unscaled velocity profile (figure.5(e)) starts from non-zero velocity and the scaled velocity profiles (fig.5(f) and figure.5(g)) also starts from the same point. Hence unlike constant scaling [1], non-linear scaling proposed here does not result in any discontinuity in the velocity space. Figure.5(h) and figure.5(i) shows the satisfaction of permanent contact and no-slip constraint for the velocity and acceleration profile obtained with scaling function 2.

#### B. Trajectory Replanning

In this example we show how unstable portions of the trajectory is re-planned while keeping the stable portions intact. Figure.6(a) shows an initial planned path ( $\chi(u)$ ) for which at some places no  $s_{min}$  exist (figure.6(b)). The portion in black represents the unstable portion of the trajectory while the portion in red denotes the stable portion of the trajectory. So keeping the stable portion intact a new trajectory ( $\chi_r(u)$ ) is planned (shown in blue). So the final

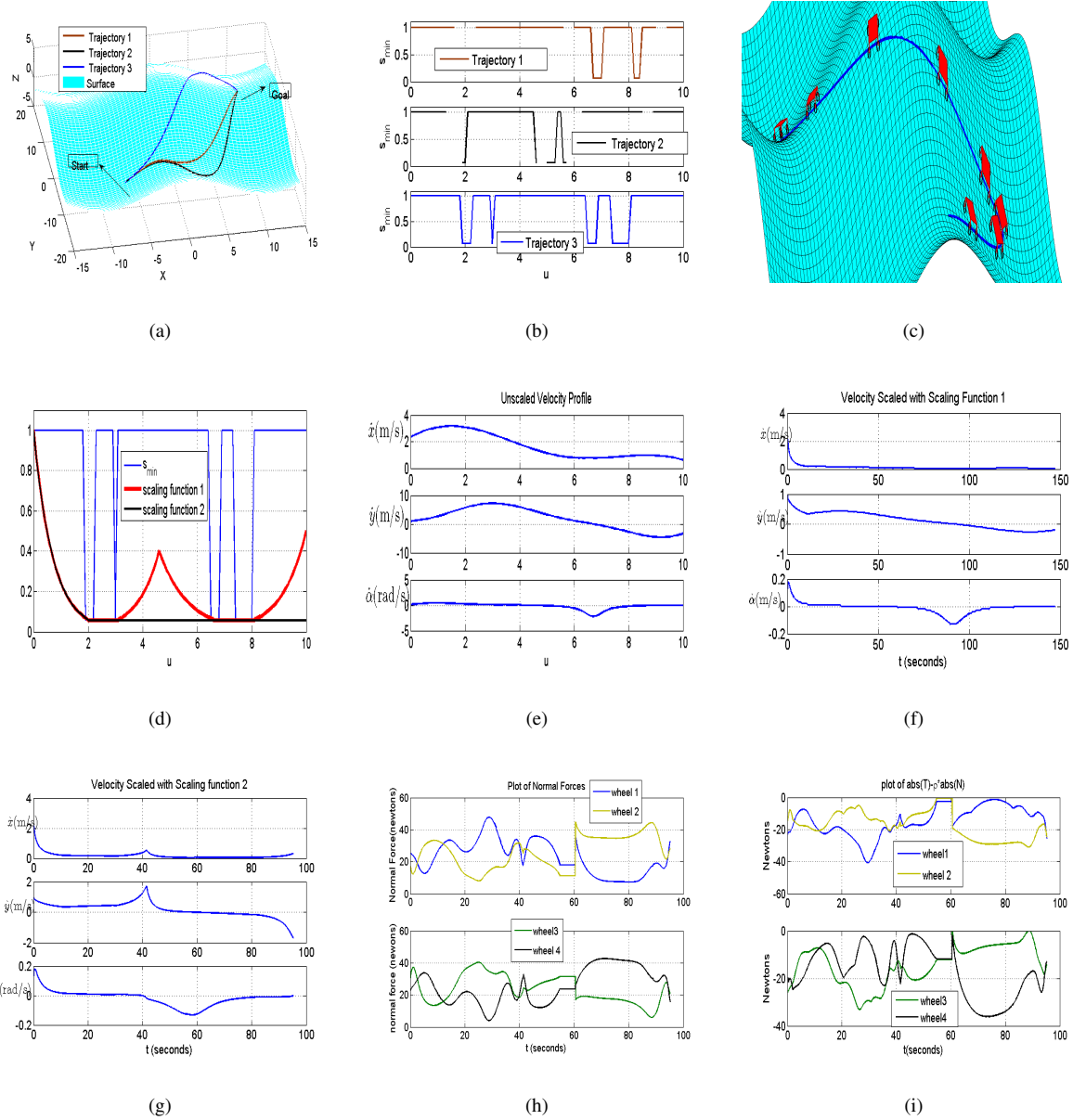


Fig. 5. (a) Multiple Trajectories between a start and goal location. (b) Variation of  $s_{min}$  along the obtained trajectories. (c) Vehicle's Evolution along stable trajectory 4. (d) Scaling function for the stable trajectory 4. (e) Unscaled Velocity Profile for trajectory 4. (f) Velocity profile for trajectory 4 with scaling function 1. (g) Scaled velocity for trajectory 4 with scaling function 2. (h) Satisfaction of (19). (i) Satisfaction of (20)

trajectory from start to the goal will be a concatenation of  $\chi(u)$  and  $\chi_r(u)$ . Let it be called as  $\chi_c(u)$ . Figure.6(b) shows the plot of  $s_{min}$  for the combined trajectory and the scaling function constructed corresponding to it. The scaling function is defined in the following manner:

$$\frac{du}{dt} = \begin{cases} 1, & 0 \leq u \leq 5 \\ 468.71e^{-1.23u}, & 5 \leq u \leq 7.3 \\ \frac{1}{17}, & 7.3 \leq u \leq 10 \end{cases} \quad (26)$$

It can be noted from this definition that for the first 5 sec. the  $\frac{du}{dt}$  is constant at 1 which means that for the first 5 sec. the scaled and unscaled velocity profile are the same. This can be confirmed from figure.6(e). This shows how velocity and acceleration profile can be scaled down

midway while executing a trajectory. This is a very important feature of the proposed non-linear scaling method which was not possible with constant scaling [1]. Figure.6(d) shows the scaled velocity profile and figure.6(f) and figure.6(g) confirms the satisfaction of permanent contact and no-slip constraint with the scaled velocity and acceleration profile.

### C. Comparison of quality of parametric trajectories with that obtained from smpling based motion planners

The computational gain of parametric trajectory generation over sampling based motion planners are well known but the former can possibly introduce sub-optimality in the quality of the path. Since the addressed problem is of uneven terrain trajectory generation, quality of the path is measured by

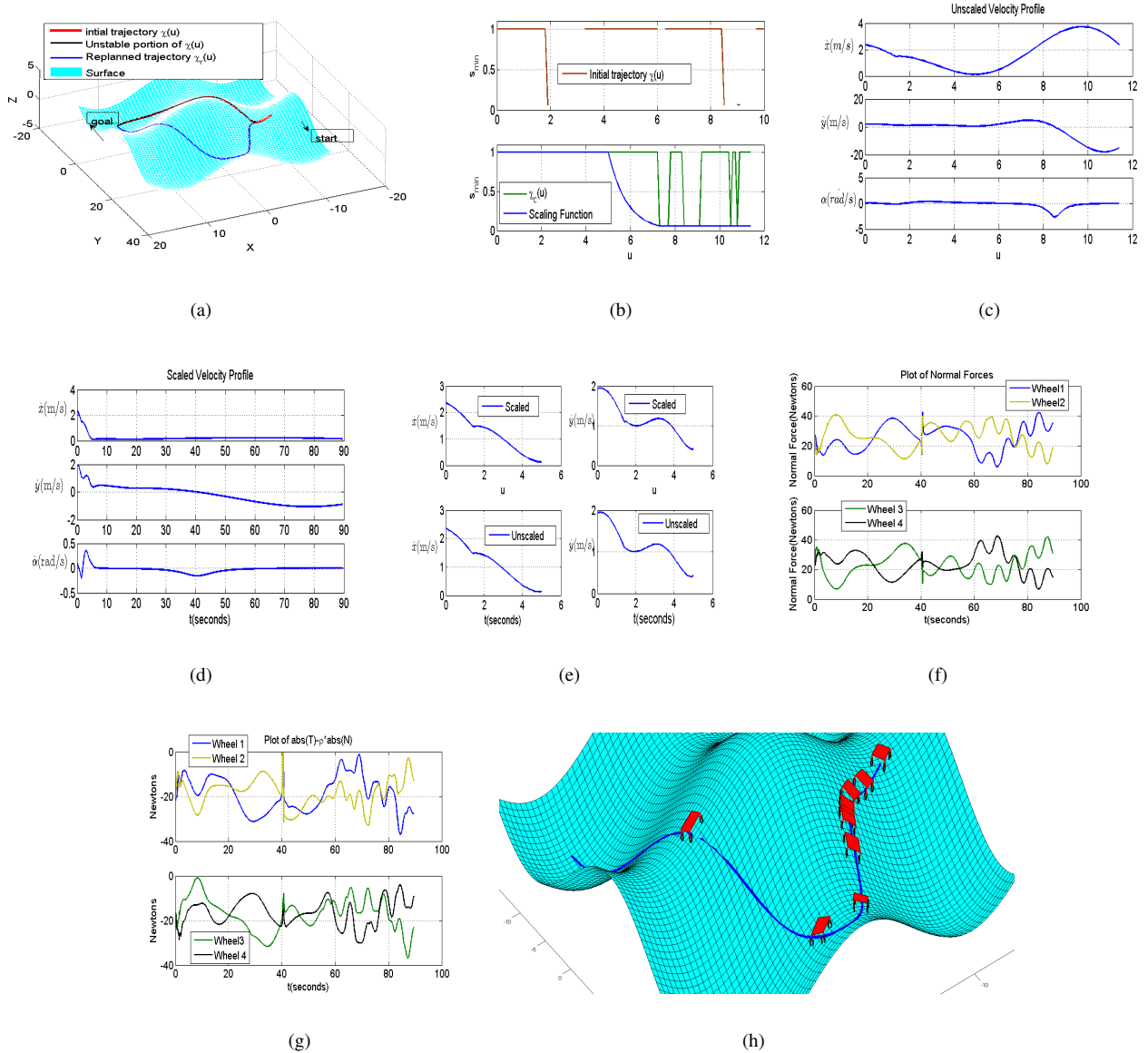


Fig. 6. (a) Trajectory Re-planning. (b) First half shows  $s_{min}$  variation along initial trajectory  $\chi(u)$ . Second half shows improvement of  $s_{min}$  along the combined trajectory  $\chi_c(u)$ . (c) Unscaled Velocity Profile of  $\chi_c(u)$ . (d) Scaled velocity profile of  $\chi_c(u)$ . (e) Confirmation that for the first 5 seconds the scaled and unscaled velocity profile are the same. This shows it is possible to apply non-linear scaling transformation midway while executing a trajectory. (f) Satisfaction of (19). (g) Satisfaction of (20). (h) Vehicle's evolution along  $\chi_c(u)$

the stability metric FAC proposed in [5]. FAC denotes the measure of the space of feasible accelerations for the current velocity profile which satisfies (19) and (20). It was shown to be a better metric than the more commonly used Tip-Over metric [20]. Table I summarises the FAC comparison for five terrains averaged over 20 different paths with different start and goal location. Optimization based trajectory have on an average FAC 0.9 times that of sampling based planner which means that the quality of parametric trajectories are comparable to that obtained from sampling based motion planner involving exhaustive search.

Figure.7 shows the number of iterations required for optimization to converge, averaged for 200 paths with particular  $\zeta_f$  and different start and goal location . Arbitrary initial

guess was given to the optimization routine and the tolerance for the error was kept as  $10^{-6}$ . The number of iterations taken are comparable to that reported in [3]. For every path produced by the optimization,  $s_{min}$  needs to be evaluated along the points on the trajectory. However resolution of the search space for  $s_{min}$  is not critical and the conclusion about its existence can be made by evaluating 10 to 20 scale factors. In our case we evaluate 20 values in the interval (1,80) as possible  $s_{min}$  and if none of them leads to the satisfaction of (19) and (20) at any particular point, we conclude that  $s_{min}$  does not exist for that point. The optimization then produces a new trajectory through the replanning framework and the process is repeated till a completely stable trajectory from start to the goal has been constructed. The total time taken for

each path comes out to be 3-5 seconds while for a sampling based planner it is usually around 30 min, primarily because the dimensional of the search space will be 2 (linear and angular acceleration) and the resolution of the search space will have a telling effect on the output. Moreover for each point in the linear and angular acceleration search space the constraints (19) and (20) would have to be evaluated and for all those satisfying them, vehicle forward evolution would have to be computed through numerical integration. The time taken by the optimization and for the  $s_{min}$  search can be reduced significantly by using C++ platform or by parallelizing the computation in a GPU.

TABLE I

PATH QUALITY COMPARISON BETWEEN SAMPLING BASED PLANNER AND PROPOSED OPTIMIZATION BASED TRAJECTORY GENERATION

| Terrain   | $FAC(sampling)$ | $FAC(optimization)$ |
|-----------|-----------------|---------------------|
| Terrain 1 | 0.67            | 0.56                |
| Terrain 2 | 0.62            | 0.59                |
| Terrain 3 | 0.59            | 0.53                |
| Terrain 4 | 0.75            | 0.67                |
| Terrain 5 | 0.81            | 0.72                |

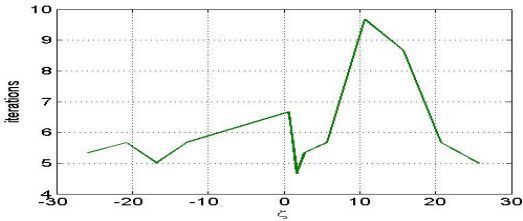


Fig. 7. Number of Iterations required for optimization to converge plotted as a function of  $\zeta$

## VII. APPENDIX

$$p_i = k_1 l_i \sin \alpha - k_2 l_i \cos \alpha \quad (27)$$

$$q_i = k_1 l_i \cos \alpha + k_2 l_i \sin \alpha \quad (28)$$

$$H_1 = k_1 x - k_1 w \sin \alpha - k_1 h \cos \alpha + k_2 y - k_2 h \sin \alpha + k_2 w \cos \alpha + k_3 + l_1 \quad (29)$$

$$H_2 = k_1 x + w(k_2 \cos \alpha - k_1 \sin \alpha) + h(k_1 \cos \alpha + k_2 \sin \alpha) + k_3 + l_2 + k_2 y \quad (30)$$

$$H_3 = k_1 x + w(k_1 \sin \alpha - k_2 \cos \alpha) + h(k_1 \cos \alpha + k_2 \sin \alpha) + k_3 + l_2 + k_2 y \quad (31)$$

$$H_4 = k_1 x + w(k_1 \sin \alpha - k_2 \sin \alpha) + h(-k_1 \cos \alpha - k_2 \sin \alpha) + k_3 + l_2 + k_2 y \quad (32)$$

## VIII. CONCLUSIONS AND FUTURE WORK

### A. Conclusions

We presented a novel framework for trajectory generation capable of handling strict stability constraints. A rigorous analysis of the challenges and requirement for uneven terrain trajectory generation was presented. A relatively simple framework was proposed which owes great deal of its simplicity to the fact that it is possible to obtain trajectories without worrying about complicated vehicle dynamic constraints and then apply non-linear scaling transformation to satisfy the stability constraints. Trajectory replanning was embedded in the framework for trajectories which cannot be modified to satisfy the stability constraints.

### B. Future Work

The performance of the planner with incomplete information about the terrain and vehicle dynamics is currently studied. A probabilistic framework which depends upon terrain information and vehicle's 3D states can be developed to handle noises in terrain and vehicle dynamics information. Implementing the planner on outdoor robot platform is also part of the future work.

## REFERENCES

- [1] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 106, no. 1, pp. 102106, Mar. 1984
- [2] Imran Waheed, Reza Fotouhi. Trajectory and temporal planning of a wheeled mobile robot on an uneven surface in *Robotica* Vol 27 Issue 4 pp 481-498
- [3] Thomas M. Howard, Alonzo Kelly. Optimal Rough Terrain Trajectory Generation for Wheeled Mobile Robots. in *IJRR* Vol 26 No 2 Feb. 2007
- [4] J.V.Miro, G.Dumonteil, Ch.Beck, G.Dissanayake, "Kino-dynamic metric to plan stable paths on rough terrains," in Proc. *IROS 2010*, Taiwan pp 294-299.
- [5] Arun.K.Singh, K.Madhava Krishna and Vijay.Eathakota. "Planning Stable Trajectory on Uneven Terrain based on Feasible Acceleration Count," in Proc *CDC 2011* Miami Florida, pp-6373-6379
- [6] Zvi.Shiller, Yu-Rwei Gwo, "Dynamic Motion Planning of Autonomous Vehicles", in *IEEE Trans.on Robotics and Automation* Vol 7 No.2 pp 241-249, April 1991
- [7] Cherif.M, "Motion Planning for all terrain vehicles: a physical modeling approach for coping with dynamic and contact interaction constraints" in *IEEE Transactions on Robotics and Automation*, Volume 15 issue 2 pp 202-218, April 1999
- [8] Tamer Inanc, Raktim Bhattacharya, Mehmet K Muezzinoglu. "Use of parametric approximation in Real-Time Non linear Trajectory Generation". in Proc *CDC 2005* pp 6808-6813.
- [9] Yi Guo, Yi Long, Weihua Sheng. "Global Trajectory Generation for Nonholonomic Robots in Dynamic Environments". in Proc *ICRA 2007* pp 1324-1329
- [10] H.Delingette, M.Hebert, K.Ikeuchi, "Trajectory Generation with Curvature constraints based on Energy Minimization". in Proc. *IROS 1991* Osaka Japan pp 206-211
- [11] N. Chakraborty, A. Ghosal, "Kinematics of wheeled mobile robots on uneven terrain." *Mechanism and Machine theory*, Vol. 39, No.12, pp. 1273-1287, 2004
- [12] Tarokh.M and G.McDermott, "Kinematics Modeling and Analysis of Articulated Rovers," *IEEE Trans. Robotics*, vol. 21, No. 4, 539- 553, 2005.
- [13] Joseph Auchter and Carl Moore., "Off-Road Robot Modeling with Dextrous Manipulation Kinematics." *ICRA*, pp 2313-2318, May 2008
- [14] Amenti.N, M.Bern and M.Kamysselis, "A new Voronoi-based surface reconstruction algorithm," in Proc of *SIGGRAPH 1998* pp-415-21
- [15] Bajaj.C.L, F.Bernardini and G.Xu, "Automatic reconstruction of surfaces and scalar fields through 3D scans" in *SIGGRAPH 1995* pp 109-18
- [16] A.Prokos, G.Karras, E.Petsa. "Automatic 3D surface reconstruction by combining stereo vision with the slit-scanner approach" in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVIII, Part 5 Commission V Symposium, Newcastle upon Tyne, UK. 2010
- [17] Moshe.P.Mann., Zvi.Shiller. Dynamic Stability of a Rocker Bogie Vehicle: Longitudinal Motion in Proc of *ICRA*, Barcelona, Spain 205 pp 861-866
- [18] J.Jones., N.P.Ihrampetakis., A.C.Piigi. Computation of the Generalized Inverse of a Rational Matrix via Maple and Applications in *International Symposium on Computer Aided Control System Design* Dearborn, MI 1996 pp 495-500
- [19] Paolo Gallina., Alessandro Gasperatto. A technique to analytically formulate and to solve the 2 Dimensional Constrained Trajectory Planning Problem for a Mobile Robot. in *Journal of Intelligent and Robotic Systems* Vol 27 No 3, pp 237-262
- [20] D.A .Rey and E.G.Papadopoulos, "The force angle stability measure of tip-over stability margin for mobile manipulatorion", in *Vehicle System Dynamics*, vol 32 pp29-48 2000.