The 19th Conference of Mechanical Engineering Network of Thailand
19-21 October 2005, Phuket, Thailand

Motion Planning for Mobile Robots under Uncertainties using Linear Control Uncertainty Field

Amnart Kanarat [1]* Robert Sturges[2]
[1] Department of Mechanical Engineering, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang, Ladkrabang, Bangkok 10520
[2]Department of Mechanical Engineering, School of Engineering,
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA
*E-mail: kkamnart@kmitl.ac.th

Abstract
This paper addresses the problem of motion planning for nonholonomic mobile robots working in extreme environments, for example, desert, forest, and mine. In such environments, the mobile robots are highly subject to external disturbances, which introduce control and sensory uncertainties. The complexity of this motion planning problem arises when both nonholonomic constraints and uncertainties must be taken into account simultaneously. In this work, we seek the most robust path/motion respecting nonholonomic constraints for mobile robots in the presence of control and sensory uncertainty. A new motion planner based on a Linear Control Uncertainty Field (LCUF) approach is proposed. In this approach, the allowable control uncertainty for every configuration of a mobile robot in a free workspace is computed, and then searched by an optimization method for the most robust path connecting given initial and goal configurations in the workspace. The path obtained from this motion planner is proved to be optimal, and the minimum LCU of the path can be used to estimate how much control uncertainty is allowed while avoiding collisions.

Keywords: Nonholonomic, Motion Planning, Mobile Robot, Uncertainty

1. Introduction

Autonomous mobile robots are being developed and deployed in many real-world applications, for example, underground mining, military surveillance, and even space exploration. In those applications, the mobile robots often operate in harsh working environments, where the robots tend to be more susceptible to disturbances from the environment. The effect of the uncertainties on the performance of the robot navigation becomes critical when it works in cluttered or tight environments, i.e., where the robot size is comparable to the size of its working space. For the mobile robot to effectively navigate itself through such environment under uncertainties, its navigation system must take advantage of the information about the environment geometry as well as the mobile robot geometry to minimize the effect of uncertainties.

The goal of this research is to construct a motion planning method capable of finding a path that is most tolerant to uncertainties. A large number of motion planning techniques for nonholonomic mobile robots (e.g., wheeled or tracked mobile robots) have been proposed using various approaches. Optimal control approaches, based on Pontryagin's Maximum Principle, have been applied to find optimal controls that minimize some objective functions (e.g., time, distance, expended energy, and etc.) subject to some constraints such as actuator limitations [1].

Nonlinear control techniques include differential-geometric and differential-algebraic techniques [2], geometric phase methods [3], and control parameterization approaches [4]. All of these studies focus on determining open loop controls which steer a nonholonomic system from an initial state to a final state over finite time in an unobstructed workspace. Laumond et al. [5] applied the results from optimal control and nonlinear control techniques in developing motion planners for nonholonomic systems in the presence of obstacles. Mirtich and Canny [6] have proposed a unique approach in planning a path for a car-like robot moving among obstacles using maximal clearance skeletons. However, none of the approaches previously mentioned takes into account the effect of uncertainties, which happen in real applications.

In the past decade, a group of researchers have started to incorporate the effect of uncertainties into the motion planning problem for nonholonomic systems. Among the pioneer researches is Jacobs and Canny's work [7]. They developed the concept of δ-robustness as a means to measure the degree of robustness of a planned path. The application of probability theory was also introduced when Timcenko and Allen [8] proposed a motion planning method for mobile robots in the presence of sensory and control uncertainties.

Recently, a work by Lambert and Le Fort-Piat [9] has shown a method in developing a motion planning for mobile robots in the presence of bounded uncertainties by introducing the concept of a security margin. However, the authors only used the concept of security margin as a means to check robustness of a pre-planned path generated by an A* search method but not to plan a path.

It is clear that none of the previous works studies directly how the degree of uncertainties dictates the shape of a planned path in a given environment. Moreover, the previous works cannot provide answers to how much uncertainty in a mobile robot is allowed to navigate safely through a given environment on a given path, and to what the best possible path is for a given environment. To address the shortcoming of the existing works, we have developed a novel motion planning algorithm for nonholonomic mobile robots under control and sensing uncertainties. The algorithm plans the most robust path that the robots must follow, so that the likelihood of collisions due to the presence of uncertainties is eliminated. We also have developed a method for determining the allowable maximum degree of control and sensing uncertainties for the mobile robots to follow a given path safely.

This paper is organized as follows. First, the mobile robot and workspace models used throughout this research are introduced. Second, the types of uncertainties in robot systems and how they can be combined are discussed. Third, the concept of combined uncertainty leading to the introduction of "Linear Control Uncertainty (LCU)" is given, and this concept is later developed into the "Liner Control Uncertainty Field (LCUF)". Then, a new type of nonholonomic motion planning based on the LCUF is proposed. Finally, the results and discussion, including the conclusion, are given.

2. Robot and Workspace Models

A kinematic model for a differentially-driven wheeled mobile robot is selected to be the robot model for both wheeled and tracked mobile robots. Equation (1) shows the system of ODEs representing the robot model.

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

$$x = [x_c, y_c, \theta]^T \text{ and } u = [v, \omega]^T$$

where x is the system state, and (x_c, y_c) are the coordinates of the origin of the robot-fixed coordinate frame *{xy}* with respect to the global coordinate frame *{XY}*, see Figure 1. The angle θ represents the robot heading angle with respect to X-axis. Control u consists of the linear velocity v and the angular velocity ω.

From (1), the controls of the mobile robot are linear and angular velocities. However, the actual controls we provide to the mobile robot are right and left wheel velocities as shown in Figure 1. The following equations can be used to computed the linear and angular velocities when the values of right and left wheel velocities, v_r and v_l, are available.

$$v = \frac{v_r + v_l}{2} \qquad \omega = \frac{v_r - v_l}{B} \quad (2)$$

where B is the robot wheel base.

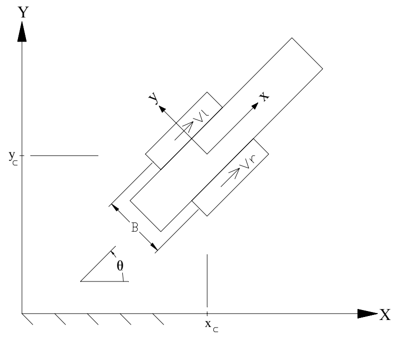

Figure 1. A mobile robot model.

Knowledge about the robot workspace in this study is assumed to be partially known. A two-dimensional map representing approximately the locations of obstacles comprises segments of straight lines forming polygons and boundaries. The segments of straight lines are represented by a set of linear equations and their end points in the global coordinate frame *{XY}*, as shown in (3) below.

$$\begin{aligned} & Y = m_i X + c_i \\ & s_i = (Xs_i, Ys_i) \\ & e_i = (Xe_i, Ye_i) \\ & i \in [1, N] \end{aligned} \quad (3)$$

where m_i = slope, c_i = Y-axis intercept
s_i = start point, e_i = end point
i = index number, N = total number of lines

A line parallel to the Y-axis will be represented by (4) with its end points.

$$x = d_i \quad (4)$$

3. Uncertainties in Robot Systems

The primary forms of uncertainties entering robot systems can be categorized into three types: model, sensory, and control uncertainties. Model uncertainty appears due to un-modeled dynamics and inaccurate parameters of the robot system. Sensory uncertainty happens from the fact that there is no perfect sensor which can measure physical quantities exactly. Therefore, sensory data obtained from sensors always have more or less built-in uncertainties. Control uncertainty can also be clearly seen from the objective of control itself. The discrepancy between a desired output and a real output of a control system gives rise to the control uncertainty.

In this research, the kinematic model (1) is used to describe the robot system. At any instance, this kinematic model posits a constraint that the robot can only move in the direction tangent to its trajectory, but cannot move sideways. This constraint seems to hold at all times for every mobile robot moving and cornering at low speed. Therefore, the model uncertainty is assumed to be zero, and only uncertainties from sensing and control are considered in this study.

Degrees of sensory and control uncertainties vary from one robot system to another. The degrees vary widely and depend heavily on types of sensors, actuators, and control algorithms used in the robot systems. The degrees of sensory and control uncertainties for each robot system are treated as two separated numbers, where the sensory uncertainty may be represented in the form of uncertainty ellipsoid and the control uncertainty may be represented in the form of a percentage of steady-state error. Each number may vary for different configurations or remain constant for the entire configuration space; however, its value must be bounded. The degrees of uncertainties can be predetermined either by means of experiments or by means of simulations. To generalize and simplify the analysis, the degrees of uncertainties from both sensory and control are combined and represented by a single number. Although the sensory and control uncertainties may have different dimensions, the combining of the two uncertainties can be accomplished through the use of a feedback control law of a robot. Let assume that the feedback control *u* in (1) is in the following form (assuming that all states are available):

$$u = f(x_c, y_c, \theta). \qquad (5)$$

Let the estimated state from the sensor measurement be $x_e = [x_c + \delta x \quad y_c + \delta y \quad \theta + \delta\theta]^T$, where δx, δy, and $\delta\theta$ are sensing errors in x_c, y_c, and θ due to sensory uncertainty. Thus, the feedback control (5) with the estimated state becomes:

$$u + \Delta u = f(x_c + \delta x, \quad y_c + \delta y, \quad \theta + \delta\theta) \qquad (6)$$

where, Δu is the feedback control error due to sensory uncertainty.

In addition, the feedback control (6) is also corrupted by the control error Δc due to control uncertainty. This control error is a scalar value and generally expressed in terms of a fraction of steady-state error to the desired set point. Thus, the actual feedback control entering the system (1) is:

$$(1 + \Delta c) \cdot (u + \Delta u) = u + [\Delta c \cdot (u + \Delta u) + \Delta u] \quad (7)$$

Notice that the expression in the closed bracket in the right hand side of equation (7) is an overall error in feedback control due to both sensory and control uncertainties. Clearly, the overall error in feedback control is a function of the feedback control *u* itself—both explicitly shown as *u* and inherently embedded in Δu. This means that if the feedback control *u* and the uncertainties (δx, δy, $\delta\theta$, and Δc) are bounded, the overall error in feedback control is also bounded. Consequently, the actual feedback control (7) is also bounded. From equation (7), the overall error in feedback control (or the overall control error) can be rewritten in the form of (8) by factoring out *u* from Δu and summing all coefficients of *u* together.

$$u + [\Delta c \cdot (u + \Delta u) + \Delta u] = u + \Delta_{err} \cdot u \qquad (8)$$

where, Δ_{err} is the fraction of the overall control error to the feedback control *u*.

In this form, the overall control error in equation (8) can be represented by the multiplication of the feedback control *u* with a single scalar value, Δ_{err}, which we call "Coefficient of Overall Control Uncertainty", or, in short, "Overall Control Uncertainty". This scalar value is a result of lumping together the consequences of both sensory and control uncertainties to feedback control. The overall control uncertainty conveniently and effectively represents the effect of both uncertainties as a whole to the closed-loop control of a mobile robot system. The value of the overall control uncertainty is zero when there are no uncertainties, and it has a finite positive real value when there are bounded uncertainties presented in the robot system. It is clear that the value of the overall control uncertainty also depends on the choice of feedback control (5); however, the discussion about choosing the most suitable (or optimal) feedback controller lies outside the scope of this study.

4. Linear Control Uncertainty Field

We have learned that the degree of uncertainties can be represented through a single number, the overall control uncertainty, and also learned that this number depends on the choice of feedback control given the same degree of uncertainties. This may result in different paths planned for different controllers selected. However, an optimal path for a given robot and environment should be invariant regardless of the choice of a robot feedback control. Therefore, by slightly modifying the concept of overall control uncertainty, the notion of Linear Control Uncertainty (LCU) is developed. The main idea is to determine (at each configuration in the environment) the maximum allowable overall control uncertainty instead and to use this number in defining the optimal path. This way the optimal path stays unchanged with respect to different feedback control schemes since the value of LCU does not depend on types of control. The LCU indicates the degree of combined uncertainties—both sensory and control uncertainties—allowed at any particular configuration in the robot workspace, such that the mobile robot can move from the current configuration for one time period without bumping into obstacles in the environment. Before we give a formal definition of the LCU, some of the basic definitions used in robot motion planning will be introduced first [10].

Definition 1 A **robot** $\mathcal{A}$ is a rigid object described as a compact (i.e. closed and bounded) subset of a Euclidean space $\mathcal{W}$, called **workspace**, represented as R^N, with *N* = 2 or 3. **Obstacles** $\mathcal{B}_i$, where *i* = 1, 2, …, N, are closed subsets of $\mathcal{W}$.

Definition 2 A **configuration** q of a robot $\mathcal{A}$ is a specification of the position and orientation of the robot-fixed coordinate frame *{xy}* with respect to the global coordinate frame *{XY}*. The **configuration space** of $\mathcal{A}$ is the space $\mathcal{C}$ of all the possible configurations of $\mathcal{A}$. The subset of $\mathcal{W}$ occupied by $\mathcal{A}$ at configuration q is denoted by $\mathcal{A}(q)$. A set of the interior points of set $\mathcal{S}$ is called the *interior* of $\mathcal{S}$, and denoted by $int(\mathcal{S})$.

Definition 3 The obstacle $\mathcal{B}_i$ in $\mathcal{W}$ maps in $\mathcal{C}$ to the region $\mathcal{CB}_i = \{ q \in \mathcal{C} \mid \mathcal{A}(q) \cap \mathcal{B}_i \neq 0 \}$.

$\mathcal{CB}_i$ is called a **C-obstacle**.

Definition 4 A **free space** is the subset of $\mathcal{C}$ defined by $\mathcal{C}_{free} = \mathcal{C} - \bigcup_{i=1}^{N} \mathcal{CB}_i$.

Definition 5 A **contact space** is the subset of $\mathcal{C}$ made of configurations at which $\mathcal{A}$ touches one or several obstacles without overlapping any, or mathematically defined as:

$$\mathcal{C}_{contact} = \{q \in \mathcal{C} \mid \mathcal{A}(q) \cap \bigcup_{i=1}^{N} \mathcal{B}_i \neq 0 \text{ and } int(\mathcal{A}(q)) \cap \bigcup_{i=1}^{N} int(\mathcal{B}_i) = 0 \}.$$

Valid space is: $\mathcal{C}_{valid} = \mathcal{C}_{free} \cup \mathcal{C}_{contact}$.

We are now ready to state the definition of the Linear Control Uncertainty (LCU).

Definition 6 (Linear Control Uncertainty) Let T be a constant time period, and $T > 0$. Given a robot $\mathcal{A}$ moving in a workspace $\mathcal{W} \in R^2$ containing obstacles $\mathcal{B}_i$, the **Linear Control Uncertainty** (LCU) of a configuration $q \in \mathcal{C}_{free}$ is a nonnegative real number that perturbs a constant nominal control V (where $V \in R$) for the actual controls v_r and v_l in (2), as defined by the following equations

$$\begin{aligned} v_r &= (1+LCU)\cdot V, \quad v_l = (1+LCU)\cdot V \\ v_r &= (1-LCU)\cdot V, \quad v_l = (1-LCU)\cdot V, \\ v_r &= (1+LCU)\cdot V, \quad v_l = (1-LCU)\cdot V \\ v_r &= (1-LCU)\cdot V, \quad v_l = (1+LCU)\cdot V \end{aligned} \tag{9}$$

such that the trajectory of system (1) starts from $x(0) = q_0 = q \in \mathcal{C}_{free}$ and ends at $x(T) = q_T \in \mathcal{C}_{free}$ with $x(T^+) = q_{T^+} \in \mathcal{C}_{contact}$, where $T^+ \in (T, \infty)$.

From the definition of LCU, there are two free variables. One is the constant nominal control V, and the other is the constant time period T. These two variables can affect the value of LCU of a free configuration in a given workspace. For example, if one decreases V and T, the value of LCU will increase, or vice versa. Although the selection of these two variables affects the value of LCU for all free configurations, the same effect applies to all free configurations. Therefore, the topology of the Linear Control Uncertainty Field, which will be discussed next, of a free configuration space are invariant to the choices of V and T. This means that the configuration possessing the higher LCU than its neighbors always possesses the higher LCU than its neighbors for different sets of V and T. In general, we choose V to be the maximum or minimum velocity of the robot wheel, and choose T to be a sampling period.

The reason to let V be the maximum or minimum velocity is to establish a lower bound for the value of LCU for any given T, so that the maximum allowable error in the closed-loop control system can be stated. The reason to let T be a sampling period is that for many practical control schemes, especially digital ones, the value of the control input to dynamic systems is held constant for one sampling period such as a zero-order hold in digital control. Further notice that the nominal control V is the same in both equations of v_r and v_l in (9), this is because, for determining LCU, the nominal trajectory or path of system (1) is chosen to be a short straight line. Also notice that the LCU is assumed to be the same for both v_r and v_l , this makes sense because both v_r and v_l, at any instant, should experience the same magnitude of perturbation, but can be different in directions, shown as plus and minus signs in (9). Intuitively, the LCU, for an arbitrary configuration, with above suggested choices of V and T can be thought of as the maximum allowable degree of perturbation to the nominal straight path, such that a robot can move from that arbitrary configuration along the perturbed path for one sampling period in a workspace without touching obstacles in the workspace.

The value of LCU at a particular configuration can be determined by two means: simulation and approximation. Here, we will consider the simulation means while the approximation means will be discussed in another paper devoted to an extension of the LCU. Given a robot, a robot workspace, obstacles, V, and T, the simulation iterates for each particular configuration by increasing LCU (starting from zero), calculating perturbed controls from equation (9), integrating system equation (1) forward in time for T seconds, and checking for a collision between a robot and obstacles. If there is no collision, the LCU is increased and another iteration is carried out. The simulation terminates when collision occurs, and the LCU just before the collision is declared as the LCU of that particular configuration.

Two types of collision detection algorithms have been used in this study. One is based on a direct implementation of linear algebra by determining all possible intersections between each pair of the edges of the robot and obstacles. The other is based on the "Separating Axis Theorem" [11]. The linear algebra-based algorithm uses less computation time than the separating-axis-theorem-based algorithm does when the obstacles in the problem are concave and must be decomposed into smaller convex obstacles for the separating-axis-theorem-based algorithm to work.

The concept of a LCU is directly applied to the motion planning problem of nonholonomic mobile robots. The idea is to find a path that connects given initial and goal configurations, while all of the configurations on the path possess high values of LCUs. To be able to determine the LCUs along a path, the path must be approximated by a series of discrete configurations. Before searching for an optimal path in a discrete configuration space, the LCUs at each discrete configuration q in a free space $\mathcal{C}_{free}$ must be computed first to create a "Linear Control Uncertainty Field (LCUF)", and the search can be conducted afterward. The definition of this field is as follows:

Definition 7 Given a robot $\mathcal{A}$ moving in a workspace $\mathcal{W} \in R^2$ containing obstacles $\mathcal{B}_i$, a **Linear Control Uncertainty Field** (LCUF) is a field of LCU for all $q \in \mathcal{C}_{free}$.

An example of a LCUF is presented in Figure 2. The figure depicts a graphical representation of a LCUF of a rectangular mobile robot (shown on the top of the figure) moving in a given workspace, a 90-degree turn passage. The magnitudes of the vectors in the figure show values of LCU at the points from which the vectors emanate. The directions of the vectors represent heading angles or orientations of the mobile robot at those points. One can notice that the value of LCU is highly sensitive to changes in the orientations of mobile robot as the magnitudes of the vectors at the same point vary greatly when there are small changes in orientations. This characteristic of the LCU poses a difficulty in searching for an optimal path through the LCUF since the values of a cost function of slightly different candidate paths can be considerably different, which gives rise to a highly nonlinear optimization problem. Nevertheless, the method used in the next section can cope with this difficulty.

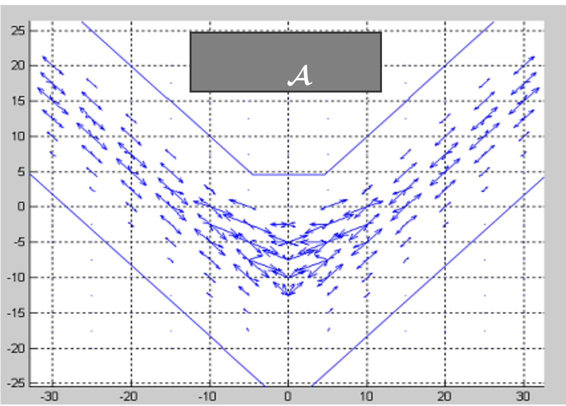

Figure 2. A Linear Control Uncertainty Field (LCUF).

4.1 Optimal Path Searching Method

For an optimal path, the system dynamic model (1) must be satisfied at all times. Additionally, the optimal path—connecting an initial configuration and a goal configuration—of the system must minimize a given cost function. In this setting, the nonholonomic motion planning problem can be converted into an optimal control problem. When one determines an optimal control, v^* and ω^*, that steer the system (1) from a given initial state to a given terminal state such that a given cost function is minimized. This way it is guaranteed that the resultant path (or trajectory) always satisfy the nonholonomic constraint.

One of the popular techniques used in optimal control problems is to transform an optimal control problem into a standard optimization problem, and then use well-established tools in optimization theory to solve the optimal control problem. This technique uses approximation-and-optimization approach, which is widely accepted, and also used in commercial codes such as POST (Program to Optimize Simulated Trajectories) written by Lockheed Martin Astronautics and NASA [12].

A motion planning problem is a constrained optimization problem by nature due to constraints in both the state of the system (x_c, y_c, and θ) and the control of the system (v_r and v_l). The constraint of the state comes directly from obstacles in a workspace, while the constraint of the control arises from the fact that the control is bounded. To convert the constrained optimization problem to an unconstrained one, we utilize the concept of penalty functions, where we penalize the state and control constraint violation (represented by terms *col* and $\beta(u)$, respectively, in cost function(10)). In addition, to make sure that the final configuration of the mobile robot terminates at the goal configuration, the cost function also includes a quadratic function of the discrepancy between the final and goal configurations, $(x_N - x_g)^2$. Ultimately, the most desired path should comprise configurations possessing the greatest LCUs. With all these quantities, the cost function can be explicitly written as follows:

$$F(u) = -\min_{i \in [1,2,\ldots,N]} LCU(x_i) + \{\ \alpha(x_N - x_g)^2 + \beta(u) + col\ \}, \qquad (10)$$

where, $LCU(x_i)$ = LCU at configuration i.

α = constant positive weight.

x_N = final configuration of the robot.

x_g = goal configuration of the robot.

$\beta(u) = 0$ if no control constraint violation occurs.

$= 1000 \cdot (\|u\|_\infty)^2$ if control violation occurs

col = 0 if no collision occurs.

= 1000 · $\min LCU(x_i)$ if collision occurs.

The cost function F is a function of control u, which can be easily transformed into a series of configurations x_i (state x_i and configuration q_i are equivalent and can be used interchangeably) by simulating the system (1) to impose the nonholonomic constraint. The LCU of each x_i is varied along the path produced by the control u. The minimum one, $min\ LCU(x_i)$, is selected and used in the cost function as a minimum LCU of that path, *min LCU*. In this way, we can guarantee that the mobile robot with control uncertainty lower than this minimum value is able to traverse along the path safely so long as the mobile robot follows the path accurately. The weight, α, is used for controlling the characteristic of the end point x_N of an optimal path, for example, one can set α with a large positive number to find the optimal path which ends closer to the given goal configuration x_g. However, α must be carefully chosen such that the term $\alpha(x_N - x_g)^2$ will not dominate the term $min\ LCU(x_i)$.

The other form of cost function that we also consider is an integral form, which is the most common in typical optimal control problems. That cost function can be explicitly written as follows:

$$G(u) = -\sum_{i=1}^{N} LCU(x_i) + \{\ \alpha(x_N - x_g)^2 + \beta(u) + col\ \}, \qquad (11)$$

$$i \in [1, 2, \ldots, N]$$

where, $\sum_{i=1}^{N} LCU(x_i)$ = the sum of LCUs at each configuration along a path.

col = 0 if no collision occurs.

= $1000 \cdot \sum_{i=1}^{N} LCU(x_i)$ if collision occurs.

The differences between the two cost functions, (10) and (11), are only the first and the last terms of both cost functions. However, using the penalty functions such as col and $\beta(u)$ in (10) and (11) introduces a discontinuity to the cost function. The well-known direct search methods called the "Nelder-Mead simplex method" [13] is selected as the optimization method because it is a very effective direct search method for a multidimensional unconstrained nonlinear minimization problem having non-smooth cost function.

5. Results and Discussions

The generalized rectangular robot and the workspace, as shown in Figure 2, are chosen to be a case study. The sampling period T and the maximum wheel velocity V of the robot for the LCU calculation are 0.5 second and 1 unit/second, respectively. The goal is to find two optimal paths that minimize the cost function (10) and (11) with the same $\alpha = 0.01$, and then compare the two paths. The controls, v and ω, are bounded within [-1,1] unit/second and [-1,1] rad/second, respectively. Both controls are approximated by two piecewise constant functions, and each one is uniformly divided into 60 equal intervals, $N = 60$. Hence, we have a total of 120 variables to optimize. The final time is specified to be 60 seconds, $T_f = 60$. Given initial and goal configurations are $[-25,11,-\pi/4]^T$ and $[25,11,\pi/4]^T$, respectively.

The search program ran twice (one for each cost function) on a personal computer with a 1.7 GHz processor, and took nearly 96 hours to search for the optimal path for each cost function. The minimum LCU path and the integral LCU path in Figure 3 correspond to the cost function (10) and (11), respectively. The minimum value of the LCUs along the minimum LCU path is 2.1 or 210%, which means that the robot must have overall control uncertainty in its feedback control less than or equal to 210% to traverse along the path safely with a linear velocity less than 1 unit/second, provided that the robot follows the path accurately. The minimum value of the LCUs along the integral LCU path, on the other hand, is only 1.6 or 160%.

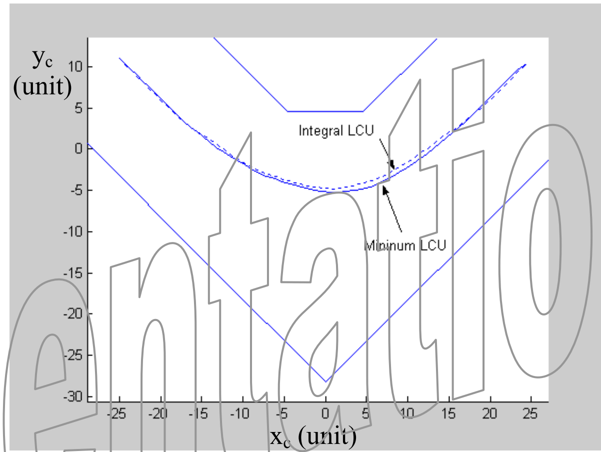

Figure 3. Optimal paths for different cost functions.

6. Conclusion

The notion of "Linear Control Uncertainty (LCU)" is proposed as a measurement, for each configuration in the configuration space, indicating how much uncertainty tolerance that the configuration has compared to its neighbor. A field of LCUs in a workspace, which we called the "Linear Control Uncertainty Field (LCUF)", can be searched by the proposed search method for an optimal path lying within the workspace. This search method, called "LCUF-based motion planning", is based on the approximation-and-optimization approach, which is often used in solving for optimal trajectories in optimal control problems. The search method yields a global optimal path with respect to the minimum LCU cost function and the integral LCU cost function.

The minimum LCU cost function yields better results, and, therefore, is favorable over the integral LCU cost function in this study. The minimum LCU of the optimal path can be used as an estimated number to determine how much control uncertainty is allowed in a robot system. In term of the applications of the proposed method, although we have only shown the application of our proposed motion planning method to solve a two-dimensional nonholonomic motion planning problem, the method, in fact, can be readily extended to solve a three-dimensional one.

Acknowledgements

The authors would like to thank Dr. Eugene Cliff from the department of aerospace and ocean engineering, Virginia Tech, for supplying parts of the optimization code.

References

[1] Alexander, J.C., et. al (1998), "Shortest distance paths for wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 5, pp. 657-662.
[2] Lafferriere, G. and Sussmann, H.J. (1993), "A differential geometric approach to motion planning," in *Nonholonomic Motion Planning*, Kluwer Academic Publishers, pp. 235-270.
[3] Reyhanoglu, M., et. al (1993), "Motion planning for nonholonomic dynamic systems", in *Nonholonomic Motion Planning*, Kluwer Academic Publishers, pp. 201-234.
[4] Murray, R.M. and Sastry, S.S. (1993), "Nonholonomic motion planning: steering using sinusoids," *IEEE Transaction on Automatic Control*, Vol. 38, No. 5, pp. 700-716.
[5] Laumond, J-P., et. al (1994), "A Motion Planner for Nonholonomic Mobile Robots," *IEEE Transactions on Robotics and Automation*, Vol. 10, No.5, pp. 577-593.
[6] Mirtich, B. and Canny, J. (1992), "Using Skeletons for Nonholonomic Path Planning among Obstacles," *Proc. of the 1992 IEEE Int. Conf. on Robotics and Automation*, France, pp. 2533-2540.
[7] Jacobs, P. and Canny, J. (1993), "Planning smooth paths for mobile robots," *Nonholonomic Motion Planning*, Kluwer Academic Publishers, pp. 271-342.
[8] Timcenko, A. and Allen, P. (1994), "Probability – Driven Motion Planning for Mobile Robots," *Proc. of the 1994 IEEE Int. Conf. on Robotics and Automation*, San Diego, pp. 2784-2789.
[9] Lambert, A. and Le Fort-Piat, N. (2000), "Safe Task Planning Integrating Uncertainties and Local Maps Federations," *The Intl. Journal of Robotics Research*, Vol. 19, No. 6, pp. 597-611.
[10] Latombe, J-C. (1991), *Robot Motion Planning*, Kluwer Academic Publishers, Massachusetts.
[11] Gottschalk, S. (2000), Collision Query using Oriented Bounding Boxes, Ph.D. thesis, University of North Carolina at Chapel Hill, USA.
[12] Brauer, G.L., Cornick, D.E., and Stevenson, R. (1977), "Capabilities and Applications of the Program to Optimize Simulated Trajectories," NASA CR-2770.
[13] Lagarias, J.C., Reeds, J.A., Wright, M.H., and Wright, P.E. (1998), "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," *SIAM Journal of Optimization*, Vol. 9, No. 1, pp. 112-147.