Robotica (2009) volume 27, pp. 567–577. © 2008 Cambridge University Press
doi:10.1017/S0263574708004955 Printed in the United Kingdom

Skiing robot - design, control, and navigation in unstructured environment

Leon Lahajnar*, Andrej Kos and Bojan Nemec

Robotics Laboratory, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

(Received in Final Form: July 9, 2008. First published online: August 15, 2008)

SUMMARY
The paper describes a skiing robot that is capable of skiing autonomously on a ski slope. The robot uses carving skiing technique. Based on a complex sensory system it is capable of autonomously navigating on the ski slope, avoiding obstacles, and maintaining a stable position during skiing on an unknown ski slope. The robot was tested using simulation in a virtual reality environment as well as on a ski slope.

KEYWORDS: Robot design; Hierarchial control; Motion planning; Autonomous motion; Virtual reality.

1. Introduction

Research in robotics has recently focused on humanoid and service robots where the final goal is to imitate human behavior and thus make a robot a useful assistant in everyday life. As a consequence, researchers study a variety of tasks performed by humans and are trying to accomplish these tasks with a humanoid or service robot. Although some tasks seem to be less appropriate for future use of humanoid robots, e.g., mountain climbing, riding bicycle, skating, etc., they are of crucial importance for an understanding of human motion and in the development of algorithms for autonomous motion of robots in unstructured environments. One such application is also alpine skiing.[10] There were only few previous attempts to develope a robotic skier.[6] In most of the cases the researchers used a humanoid robot to imitate some specific motion related to alpine skiing, but they completely ignored the problem of maintaining the stability and navigation on an unknown and unstructured ski slope.[11] Our work focuses on these two previously ignored problems. We designed a special 3-DOF (degrees of freedom) robot dedicated for skiing using the carving technique. A complex sensory system and the control system that assure the robot to stabilizes on the ski slope during skiing and tracks a path marked with race gates and avoids obstacles. The paper is divided into seven sections. The second section describes the mechanical design, actuators, sensory system, and the hierarchial control system. Lower-level control algorithms are described in the third part while the fourth section describes high-level control-navigation and decision making algorithms based on a real-time environment recognition. The fifth part describes the virtual reality environment, which we built to test the developed algorithms. Experimental results and conclusions are outlined in the sixth and seventh section, respectively.

* Corresponding author. E-mail: leon.lahajnar@ijs.si

2. System Overview

Carving skis were introduced in the late 80's and became widely used in the late 90's.[11] The main difference between the carving and traditional skis is that carving skis exploits their shape to make turns and do not require skidding[13] in order to perform a turn as do traditional skis. The essential issue is that the curvature radius of carving skis is defined by the geometry of the ski. Curvature radius can be described by

$$r = \frac{\frac{L^2 \cos\theta}{4} + \frac{h^2}{\cos\theta}}{2h}, \tag{1}$$

where coefficients L and h describe the geometry of the ski as can be seen in Fig. 1. A similar equation was presented by Howe,[7] who also accounted the penetration depth of the ski for the snow. However, the penetration depth is hard to measure or to predict on a ski slope. Note that (1) assumes that the curvature cord length is approximated by the ski length L.

From the above equation it is evident that we can control ski turn radii by controlling the angle between the skis and the skiing surface.[4] Humans are able to accomplish this action in two ways.[13] Firstly, with abduction and adduction of the hip. Hip adduction and abduction causes excessive load on the skiers muscles. Hence, skiers usually perform turns by flexion and extension of knees. To balance on the ski slope, skiers also use torso movements. This kind of turn execution motivated the design of our ski robot. We designed the lower extremities of the ski robot as two artificial legs in the form of a parallelogram. The torso has only 1-DOF and enables motion in the lateral plane. The kinematics of the 3-DOF ski robot mechanism and the CAD drawing of the mechanism are presented in Fig. 2. Note that this design enables turns only if carving skis are used.

The extension of the robot legs is controlled by DC motors with a maximum torque of 0.6 Nm and gear ratio of 157. The kinematics of leg mechanism is in the form of two connected parallelograms. In order to obtain the parallelogram motion only in the vertical plane, we have to control both the knee joints of the parallelogram simultaneously. Therefore, motor torque is transmitted to both the knee joints with a gearing, that provides simultaneous movement of both the

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press

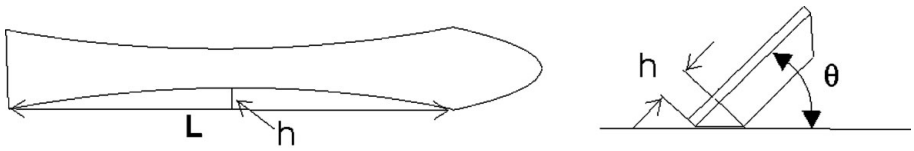


Fig. 1. Curvature radius.

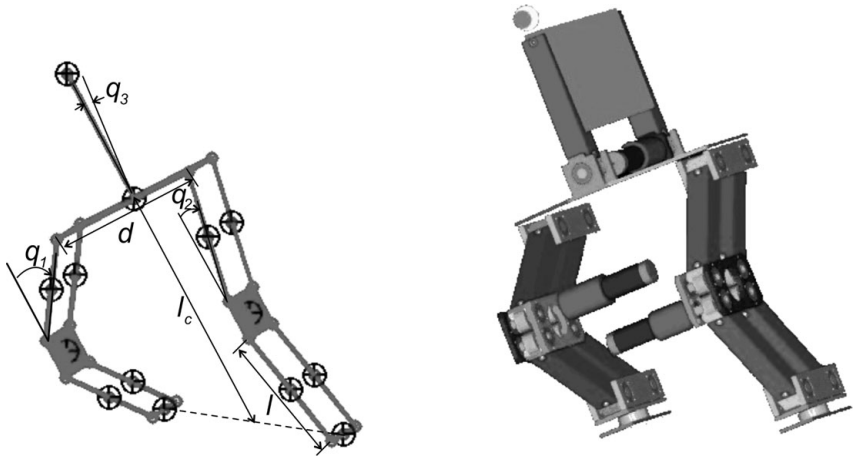


Fig. 2. Ski robot.

parallelograms. The torso is controlled by another DC motor with same characteristics and gear ratio as the knee motors. The overall weight of the robot including batteries and skis is 19 kg. The distance between the skis is 0.3 m and the overall height of fully extended robot is 0.95 m.

The robot is controlled with a hierarchically build multiprocessor computer system. The upper-level controller is used for the navigation, the vision processing and the decision making. The upper-level controller communicates with a GPS receiver and a USB based camera. The low-level controller deals with the skier stability, the joint control, and the sensory system composed of an electronic gyroscope, force sensors mounted between the skis and robot legs, and the motor position sensors. Computers communicate through Ethernet using UDP protocol. The block diagram of the robot skier control system is presented in Fig. 3.

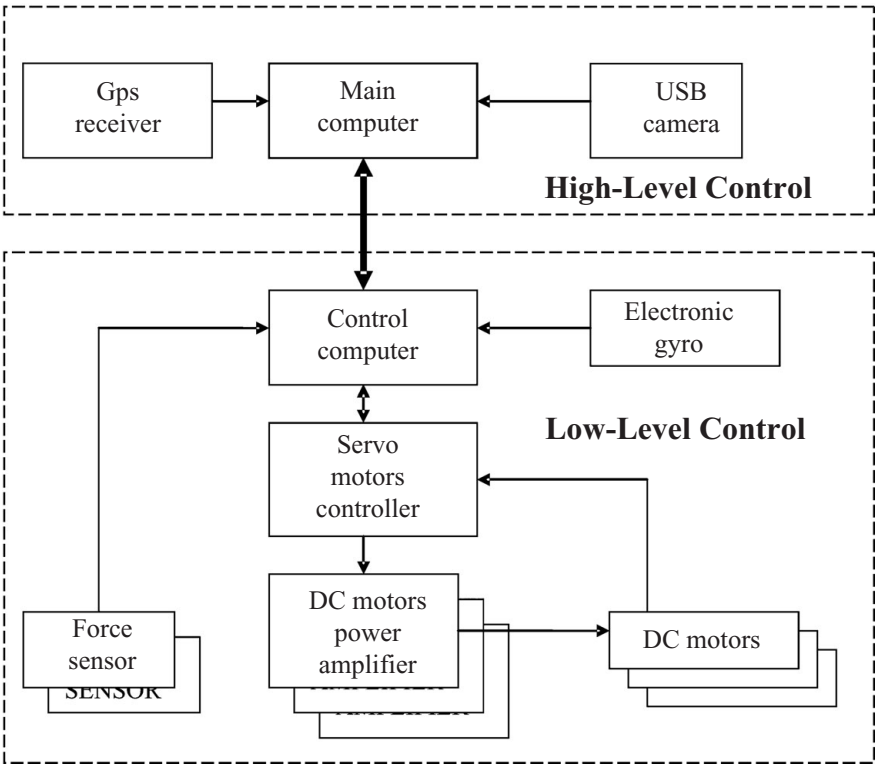


Fig. 3. Block diagram of robot skier.

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press

3. Control

The task of the robot skier control system is to assure tracking of a desired path while maintaining the stability of the mechanism in an unstructured environment, i.e., on an unknown ski slope. Additionally, the robot must be compliant to sudden shocks due to the rough ski terrain. First, the Jacobian needs to be determined, which describes the relationship between the change of curvature radius and the robot joint angle velocities. The robot controls the desired path with the inclination of the skis, which is accomplished by the flexion and the extension of the robot legs. The following equation describes the inclination angle θ of the robot with respect to the ski surface in dependence of the joint variables $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$,

$$\cos\theta = \frac{\boldsymbol{d}}{\sqrt{(2l(c_1 - c_2))^2 + \boldsymbol{d}^2}}, \tag{2}$$

where c_i denotes $\cos(\boldsymbol{q}_i)$, $\boldsymbol{d}$ is the distance between the legs, and l is length of the leg segment. Inserting (2) into (1) we obtain

$$r = \frac{\left(1/4\,\dfrac{L^2\boldsymbol{d}}{\sqrt{4\,l^2(c_1 - c_2)^2 + \boldsymbol{d}^2}} + \dfrac{h^2\sqrt{4\,l^2(c_1 - c_2)^2 + \boldsymbol{d}^2}}{\boldsymbol{d}}\right)}{2h}. \tag{3}$$

Now, the part of the Jacobian for the inclination to the ground is

$$\frac{1}{2h}[(a - b) \quad (a + b)], \tag{4}$$

where

$$a = \frac{L^2\boldsymbol{d}l^2(c_1 - c_2)s_1}{\left(4\,l^2(c_1 - c_2)^2 + \boldsymbol{d}^2\right)^{3/2}} \tag{5}$$

and

$$b = 4\,\frac{l^2h^2(c_1 - c_2)s_1}{\boldsymbol{d}\sqrt{4\,l^2(c_1 - c_2)^2 + \boldsymbol{d}^2}}. \tag{6}$$

The second DOF for the leg system is the distance l_c, defined as a distance between the midpoint of both skis and the midpoint of the robot backbone joint. By keeping this distance at appropriately constant value, we achieved the optimal manipulability of the ski robot regarding the desired task. Additionally, we used this DOF to control the robot in a lower or higher pose, just like humans adapt the skiing pose according to a specific situation. The complete Jacobian could be given in the form:

$$\begin{bmatrix} \dot{r} \\ \dot{l}_c \end{bmatrix} = \mathbf{J}\dot{q} \tag{7}$$

$$\mathbf{J} = \begin{bmatrix} \dfrac{a - b}{2h} & \dfrac{a + b}{2h} \\ -ls_1 & -ls_2 \end{bmatrix}, \tag{8}$$

where s_i denotes $\sin\left(\boldsymbol{q}_i\right)$. Using the presented kinematic control we computed the leg joint variables needed to track the desired radius. The third robot joint, which moves the torso, is controlled by the decision controller.

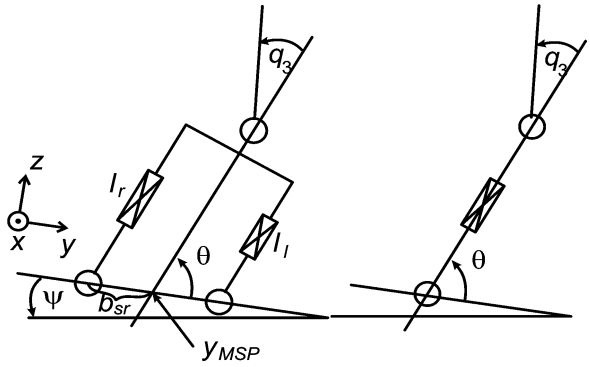


Fig. 4. Simplified model in lateral plane.

3.1. Stability of skier

The major problem in skiing is maintaining the stability in the lateral plane. In contrast, a skier is very stable in the sagittal plane, where he has enough support due to the ski length. In this direction only small variations of force are present, which appear mainly due to the change of the local inclination of the ground and of the knee forward motion. In the lateral plane we have bigger force changes, which act on a narrow support plane. Therefore, it is necessary to be focused only on lateral stability. In accordance with the previously mentioned assumptions, the model of the skier in the lateral plane can be described as shown in Fig. 4. We modeled legs as 1-DOF allowing the transitional motion in the direction of the extension of the leg.

The edging angle θ controls the curvature radius and consequently the robot's direction on the ski slope. Angle θ is controlled by the extensions (l_r, l_l) of the legs. Value l_c is the distance between the joint that actuates the trunk and the point on the surface that is in the middle between the skis. We assume that this distance is constant. In this case, the model of the skier can be represented as a double inverted pendulum.

The most stable position of the skier is when the forces are equally distributed on both skis. In this case the zero moment point (ZMP)[16] is in the middle between the legs. We will denote this position as most stable point (MSP). When the ZMP is outside of the support polygon, one ski loses the contact with the ground and the skier can fall. In the next section we present the calculation of ZMP for a serial mechanism modeled as a double inverted pendulum.

3.2. ZMP calculation on inclined surface

Suppose that the object O_i has a mass m_i at the mass center position r_i and it has the inertia tensor I_i. External forces and torques are represented by $F_{i,k}$ and $M_{i,j}$. Index k runs through all the forces acting on the i-th object, while j tracks all the torques acting on the same object. The overall rotational and translational equation of the system in an arbitrary point

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press

$p = [x_P \; y_P 0]^T$ on the plane $z = 0$ is

$$\sum_i (r_i - p) \times m_i (\ddot{r}_i + g) + \sum_i [I_i \dot{\omega}_i + \omega_i \times I_i \omega_i] - \sum_j M_j - \sum_k (s_k - p) \times F_k = M_P. \tag{9}$$

The vector s_k points to the position where the external force F_k acts and g is the gravity acceleration vector on the inclined surface and is related to the local ground inclination ψ. M_P denotes the resulting torque at the observed point. If we assume that the only external force is the radial force that works directly in the center of mass of the bodies then (9) can be rewritten as

$$\sum_i (r_i - p) \times m_i (\ddot{r}_i + g + a_{ri}) + \sum_i [I_i \dot{\omega}_i + \omega_i \times I_i \omega_i] = M_P, \tag{10}$$

where a_{ri} ($a_{ri} = [0 v^2 / R_i \; 0]^T$) is the radial acceleration of i-th object mass. R_i is the radius of the turn for each of the body segment. Because $T_i = I_i \dot{\omega}_i + \omega_i \times I_i \omega_i$ is irrelevant to an arbitrary point this leads to

$$\sum_i (r_i - p) \times m_i (\ddot{r}_i + g + a_{ri}) + T_i = M_P. \tag{11}$$

In accordance to the ZMP definition, only moment $M_P = [0 \, 0 \, M_z]^T$ acts at the point $p_{ZMP} = [x_{ZMP} \; y_{ZMP} \; 0]$. Components of the ZMP are

$$x_{ZMP} = \frac{\sum_i m_i (\ddot{z}_i + g_z) x_i - \sum_i m_i (\ddot{x}_i + g_x) z_i}{\sum_i m_i (\ddot{z}_i + g_z)} - \frac{-\sum_i (T_y)_i}{\sum_i m_i (\ddot{z}_i + g_z)} \tag{12}$$

and

$$y_{ZMP} = \frac{\sum_i m_i (\ddot{z}_i + g_z) y_i - \sum_i m_i \left(\ddot{y}_i + g_y + \frac{v^2}{R_i} \right) z_i}{\sum_i m_i (\ddot{z}_i + g_z)} - \frac{-\sum_i (T_x)_i}{\sum_i m_i (\ddot{z}_i + g_z)}. \tag{13}$$

Using the Newton–Euler formulation to derive the dynamic equations of the system, the sum of the forces $F_{0,1}$ acting on the ground and torques $M_{0,1}$ are calculated. The moment at the defined point on the ground is obtained by

$$M_P = M_{0,1} - p \times F_{0,1} \tag{14}$$

and the ZMP point is calculated by

$$x_{ZMP} = \frac{(M_{0,1})_y}{(F_{0,1})_z} y_{ZMP} = \frac{(M_{0,1})_x}{(F_{0,1})_z}. \tag{15}$$

In this way the ZMP is calculated only as a by-product of backward iteration of the Newton–Euler formulation.

As we mentioned previously, the control of the skier has three main tasks: to control the curvature radius, to assure the stability on the ski slope, and to damp the sudden force shocks due to the roughness of the ski terrain. The command variable of the curvature radius r as well as the command variable for the distance l_c is provided by the navigation module of the ski robot and will be outlined in the next section. The stability control is accomplished by the movement of the robot skier torso, which is controlled by the joint $\boldsymbol{q}_3$. In order to stabilize the robot when we cannot compensate the excessive external forces with the torso, we have to use the legs. Consequently, we cannot track the desired curvature radius. In this case the robot behavior is much like the human behavior, i.e., in order not to fall the skier changes the desired direction of the skiing. In the control algorithm we built a controller module that selects a proper action depending on the stability index and the current movement of the skier. The stability index is defined as

$$\Phi(\theta, \boldsymbol{q}_3, \psi, v) = 1 - \left(\frac{y_{MSP} - y_{ZMP}}{b_{sr}(\theta)} \right)^2, \tag{16}$$

where b_{sr} is the margin of stable region in the lateral direction and y_{MSP} is the most stable point as presented in Fig. 4.

Also in order to assure the stability in the presence of sudden and unexpected ground reaction shocks, we prescribed that the calculated ZMP should lie within the region

$$-\frac{d}{2cos(\theta)} + y_{MSP} \leq y_{ZMP} \leq \frac{d}{2cos(\theta)} + y_{MSP}. \tag{17}$$

In accordance with (16) the stability index should not fall bellow the value 0.75 in the case when we allow maximal $0.5 \, b_{sr}$ deviation of the ZMP from the MSP. In this case the force distribution between skis is 25% and 75%, respectively.

The overall control scheme is presented in Fig. 5. The local ground inclination is estimated from the gyro where a low-pass filter is applied for the elimination of unpredicted shocks. Based on the local ground inclination and the current velocity the inclination margins of the robot are calculated. When the desired turn radius cannot be achieved within the inclination margins, the commanded inclination is limited to the inclination margin. In this case the robot preserves stability but violates the desired turn radius. The difference between the commanded and the measured robot inclination is provided to the decision control block, where the sensed forces are also processed.

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press

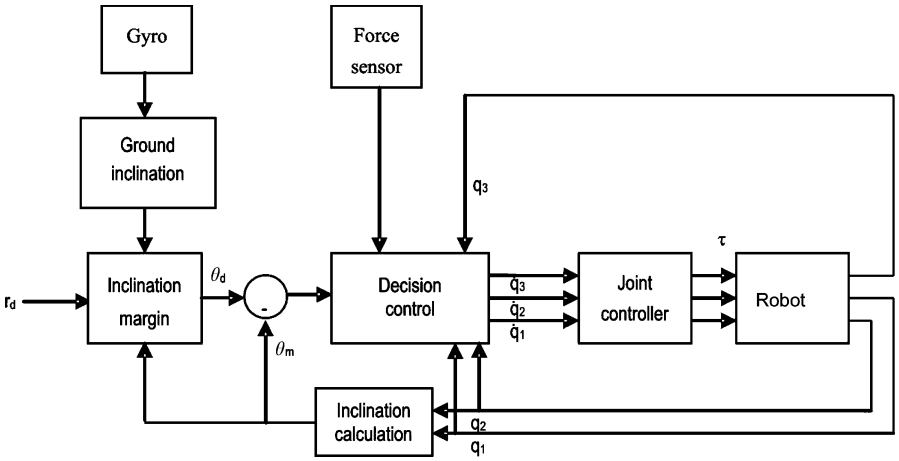


Fig. 5. Overall low-level control scheme of skiing robot.

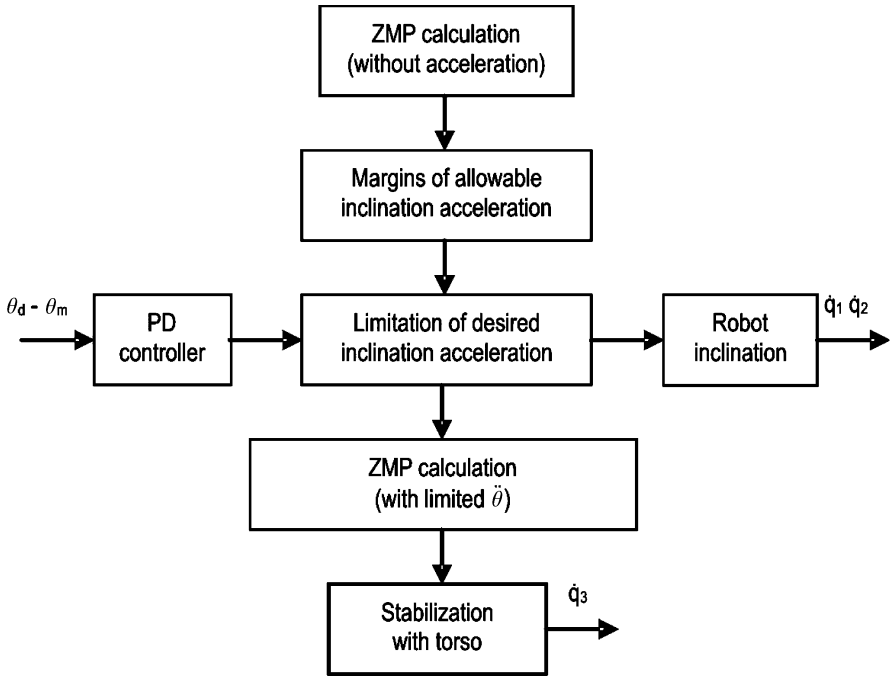


Fig. 6. Decision control.

The decision control block (Fig. 6) generates desired motor velocities using the following rules. If the stability index in the static conditions is above 0.75 then the minimal and the maximal permissible accelerations of inclination to the ground are calculated that would cause movement of the ZMP to the 0.6 b_{sr}. The control acceleration from PD controller is saturated with minimal and maximal allowable acceleration. This acceleration serves for the calculation of the angular velocity of inclination and, it corresponds to joint velocities $\dot{q}_1, \dot{q}_2$. In order to obtain the appropriate torso movement the stability index is calculated using the desired acceleration. If the stability index value is still below 0.75, the necessary acceleration $\ddot{\boldsymbol{q}}_3$ is calculated in order to stabilize the robot. In other cases the torso ($\boldsymbol{q}_3$) moves to the optimal position regarding the static stability.

As mentioned previously, the robot should be compliant to sudden shocks due to terrain irregularities. A straightforward method to reduce these shocks, uses force-sensors, is based on derivatives of measured forces. Unfortunately, the measured force signal is very noisy and derivatives of such signal cannot be used in the real-time control applications. Therefore, we accomplished active compliance based on the estimated ground reaction forces from the model. Estimated ground reaction forces are

$$F_{el} = \frac{F}{2}\left(1 - \frac{y_{ZMP}}{b_{sr}}\right), \quad F_{er} = \frac{F}{2}\left(1 + \frac{y_{ZMP}}{b_{sr}}\right). \quad (18)$$

The compliant behavior is obtained using the following control law

$$\dot{q}_1 = \begin{cases} \mathbf{K}_c(F_{el} - F_l + \sigma), & (F_{el} - F_l) < -\sigma \\ 0, & -\sigma \le (F_{el} - F_l) \le \sigma \\ \mathbf{K}_c(F_{el} - F_l - \sigma), & (F_{el} - F_l) > \sigma \end{cases} \quad (19)$$

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press

and

$$\dot{q}_2 == \begin{cases} \mathbf{K}_c(F_{er} - F_r + \sigma), & (F_{er} - F_r) < -\sigma \\ 0, & -\sigma \le (F_{er} - F_r) \le \sigma \\ \mathbf{K}_c(F_{er} - F_r - \sigma), & (F_{er} - F_r) > \sigma, \end{cases} \quad (20)$$

where F_l and F_r are the measured ground reaction forces of the left and right leg, respectively. $\mathbf{K}_c$ and σ are the chosen compliant controller gain and dead-zone, respectively.

4. Navigation

Navigation is one of the most challenging areas in the real-time mobile robot control, especially for nonholonomic mobile robots. Although global methods generally give better results, for real-time control and decision, only local methods are applicable. Baltes[1] presented a method, which is based on global planning, but enables real-time trajectory updates based on the sensory information. Many local path planning methods and obstacle avoiding algorithms are based on virtual potential fields.[8,9] Extensions to these methods have been made to incorporate robot dynamics in the local path search.[5]

Another approach was proposed by Borenstein,[2] which is based on a vector field histogram. Based on the sensor readings, the algorithm builds a function of the obstacle density. The optimal path is then calculated through the minimization of this function. This method has been extended to incorporate the robot dynamics resulting in the VFH+[14] and VFH*[15] methods. All the previously proposed methods were developed for holonomic or nonholonomic mobile robots that can control the path velocity using the drive actuators. In contrast, in the case of the skiing robot, we cannot explicitly control the velocity and the path planning algorithm must adapt to the current velocity.

As mentioned previously, the skier robot is a typical nonholonomic system and the equation of motion can be described as

$$\begin{aligned} \dot{X} &= v\cos(\varphi), \\ \dot{Y} &= v\sin(\varphi), \end{aligned} \quad (21)$$

with the restriction

$$\dot{X}\sin(\varphi) - \dot{Y}\cos(\varphi) = 0. \quad (22)$$

Here, X and Y are the robot coordinates, v is the absolute velocity and φ is the current direction of the skier projected on the plane of the ski slope. In our case, the skier direction φ depends on the current velocity and the curvature radius

$$\varphi(t) = \int \frac{v(t)}{r(t)}\,dt, \quad (23)$$

where the curvature radius r is described by. From (1), it follows that for small edging angles, θ is equal to the natural side-cut radius of unbent ski. However, experiments have shown that this equation does not describe the behavior of the ski at small edging angles satisfactorily. At low-edging angles skidding is more evident, since the ski cannot yet carve a turn into the snow. Therefore, for small edging angles ($\theta < 5^\circ$), the turning model is described with a simple relation $r = \theta^{-1}$.

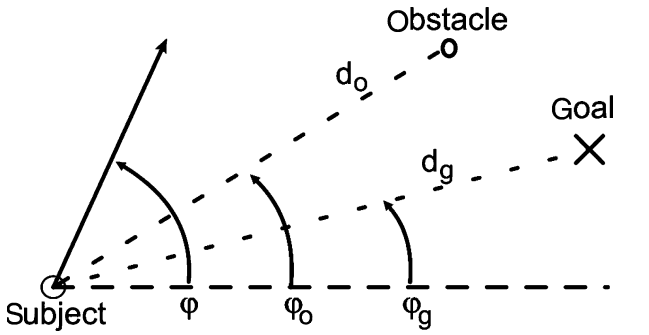


Fig. 7. Orientation of the subject in dependence of the goal and an obstacle.

Our task is to plan a path through the race gates, placed on the ski slope. Gates are both the obstacles and the local targets. The path is composed of locally generated path from one gate to the next gate. Basically, path generation can be composed of two parts. The first part is the path after the completion of the previous turn and the beginning of the next turn. During this part, the robot skier is directed towards the next gate with the direction that will enable the execution of the turn with the smallest possible radius without violating system kinematic and stability constraints. This task is accomplished using the virtual potential field, which direct the skier toward the next gate. During the second part, the skier executes the turn with the smallest possible radius. Additionally, we have to ensure the smooth transition between these two parts. Our path generation algorithm is based on the method, proposed by Fajen.[3] He has studied human navigation in a virtual reality environment. He has found out that the human body angular acceleration depends on view angles and obstacle distances. The angular acceleration can be described by

$$\begin{aligned} \ddot{\varphi} = &-b\dot{\varphi} - k_g(\varphi - \varphi_c)(e^{-c_1 d_g} + c_2) \\ &+ \sum_i k_{oi}(\varphi - \varphi_{oi})\left(e^{-c_3|\varphi - \varphi_{oi}|}\right)e^{-c_4 d_{oi}}, \end{aligned} \quad (24)$$

where φ denotes the current angle of movement of a human, φ_g is the current angle to the goal, φ_o is the current angle to the obstacle, d_g and d_{oi} are the distances between the subject and the goal and between the subject and the i-th obstacle respectively, while $b, k_g, k_{oi}, c_1, c_2, c_3$, and c_4 are the tuned parameters (see Fig. 7). Equation (24) has three components:

- $-b\dot{\varphi}$ damping component.
- $k_g(\varphi - \varphi_c)(e^{-c_1 d_g} + c_2)$ component, which directs the subject towards the goal. Scalar c_2 is used to direct the subject towards the goal at larger distances between the subject and the goal.
- $+\sum_i k_{oi}(\varphi - \varphi_{oi})(e^{-c_3|\varphi - \varphi_{oi}|})\,e^{-c_4 d_{oi}}$ component for obstacle avoidance, that rebounds the subject from nearby obstacles.

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press

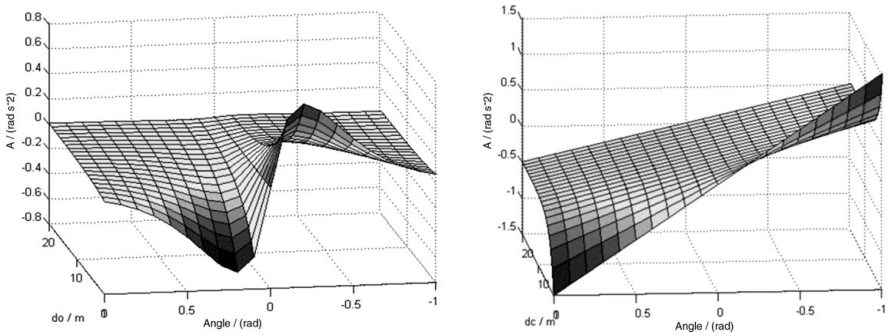


Fig. 8. Acceleration for obstacle avoidance and goal attraction in dependence of distances.

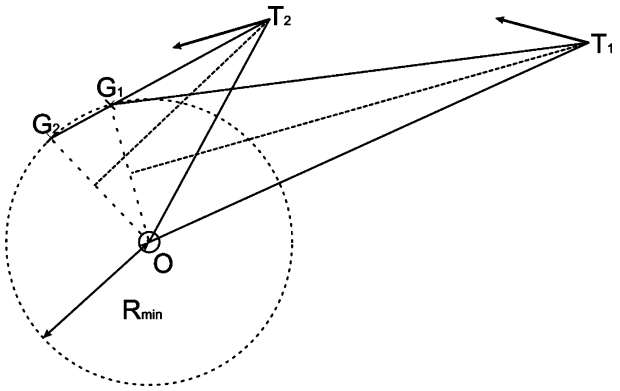


Fig. 9. Local navigation on the ski slope.

Figure 8 graphically illustrates the generated acceleration for obstacle avoidance and for acceleration towards the goal at the constant subject speed 1 ms.

Our on-line path planing procedure is illustrated in Fig. 9, where O denotes the current gate, which is the obstacle. A circle is constructed around the obstacle with radius r, which has to be bigger than the estimated smallest radius the robot is able to perform at the current velocity, estimated from the GPS system. For better radius estimation an estimation of the terrain surface in the vicinity of the gates is also needed. Unfortunately, this data is hard to obtain. At each time instance, the robot observes the scene from the point T_i. The current goal G_i, which lies on the circle with desired radius r around the obstacle, is calculated based on sensed obstacle O. When the robot touches the imaginary circle, it starts to execute the ski turn with radius r until it detects the next gate. However, this method is not optimal. An experienced skier will perform the ski turn closer to the race gates. To do this, we can shift the circle radius along the line symmetrical to the turn angle. To estimate the turn angle, we also have to estimate the position of the next race gate, which is not always possible. In such a case, we use a properly selected predefined average value for the ski turn angle.

For the navigation on the ski slope we use a vision system. The vision system has to estimate the location and the distance of the next race gate using single camera. It is assumed that the left and the right gates are marked with the blue and red fence of equal size, respectively. Therefore, the distance to the next race gate is estimated through the size of the fence. We have used edge detection technique, which detects all square objects rotated for angle $\theta + \boldsymbol{q}_3 + \psi$ in the image plane. Namely, the camera is fixed at the robot torso and the image rotates with robot inclination and the torso angle. Due to the changeable outdoor lighting conditions and due to the different view angles the fence size estimation is unreliable. We have improved the vision size estimation using filtering, which assumes, that the size of the detected object is increasing.

5. Simulation in Virtual Reality Environment

The testing of the skiing robot, control and tactic algorithms design based solely on experiments is difficult and tedious work, because it has to be accomplished outdoor on the ski slope. Therefore, a good and precise simulation tool is of beneficial. Unfortunately, the simulation is also complex task, because it requires precise model of the ski robot, the ski surface, the terrain, and also a realistic visualization of the scene. The visualization is required for the strategic control level, which is based primarily on the visual data obtained from the camera. Unfortunately, there is no simulation tool available that will meet all the mentioned requirements in real-time. Therefore, the simulation was carried out on three interconnected models. The first one is a model of the robot, second models the interaction between the skis and the surface. The last one is the visualization of the environment on which the robot is moving. Visual information is used to control the direction of skiing.

The dynamic model of the robot is used in the simulation of the robot and in the development of the control of the real system. It was built in Matlab SimMechanics. In the model we included information about mass, length, center of mass, and inertia of robot parts, which were acquired from the I-DEAS design package. The real robot has a power supply, control computer, camera, and the computer for path generation, and mechanical data of these components are also incorporated in the model. The ski robot is controlled only by changing the desired turn radius r. The real turn radius that the robot executes dependent on the dynamics, constraints, and rules that prevent the robot to fall. Data of real-turn radius are fed in the model of interaction, where the velocity data and the vector of gravity are calculated. The set of the data sent to the visualization block contains the position, the orientation, and knee and trunk angles of the ski robot, respectively. The visualization of the model is performed in Blender. The Blender package was basically designed as an environment for the design of arcade games, but has been recently

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press

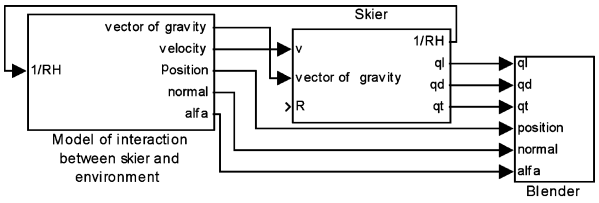


Fig. 10. Combined simulation scheme using Simulink and Blender.

recognized as an efficient tool for the robot simulation in virtual environments. The communication between the model in the Matlab Simulink and the Blender is established through UDP protocol. The robot position, its orientation, and joint angles are in sent to Blender in real-time. The same mesh of the ski slope which is used in visualization is also used in the environment model in Simulink. The top level block diagram of the simulation model is presented in Fig. 10.

The visualization allows us to reconstruct the view from the camera that is mounted on the top of the trunk. This enables us to test algorithms for the obstacle avoidance and path tracking. According to that information from the virtual camera commands for the desired turn radius are sent to the Simulink model. Figure 11 shows a scene instance taken from the robot camera and from the environment camera, which is moving along with the skier.

5.1. Simulation results

In the simulation, the robot was placed on the ski slope initial position with nonzero initial velocity. Initial velocity was 2 m/s and it was increased to around 5 m/s during the ski run. In our model we used skis with sidecut radius of 9 m. In Fig. 12 the track path of the skier together with the mesh describing the ski slope are presented.

In the virtual environment, the turn radius is controlled real-time so that the robot skis around gates. Figure 13 shows forces that are acting on the skis. Some picks are visible at the start of the simulation due to the contraction of both the legs at the initial position where ski robot can perform turns. It can be seen that during the first two turns the outer ski almost lost the contact with the surface. We can seen that with increased velocity the skier is more stable.

We also simulated the algorithm which uses the torso movement. The results are presented in Fig. 14. Note that the edging angle is in degrees divided by 20. ZMP for the torso locked in the zero position was less stable than when

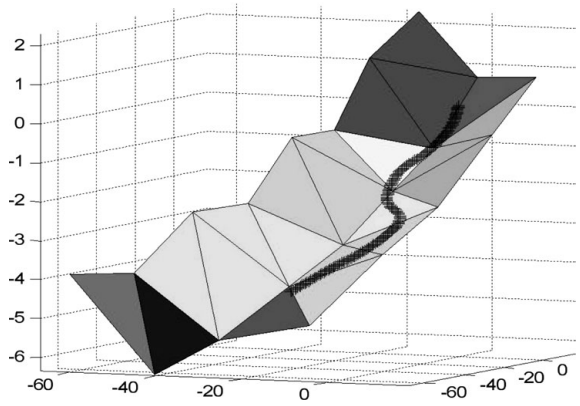


Fig. 12. Mesh of the ski slope and the simulated path of the robot skier.

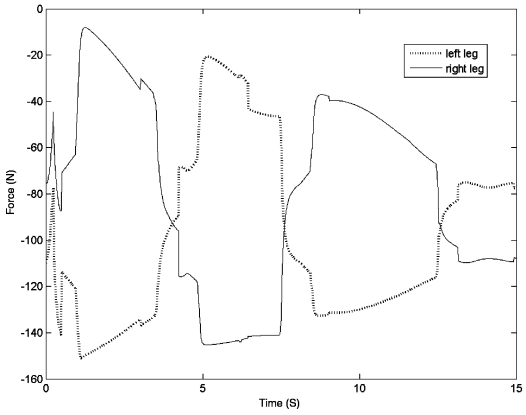


Fig. 13. Simulated ground reaction forces of both skis.

the upper body moves according to the previously mentioned rules. In both cases, performed turns were the same.

6. Experimental Results

The final test was performed on the real ski robot shown in Fig. 15. Experiments were conducted on a slope with the average inclination of 7°, where also some terrain irregularities were present.

6.1. Implementation details

The hierarchical control law was implemented using theMatlab xpcTarget toolbox with the control sampling

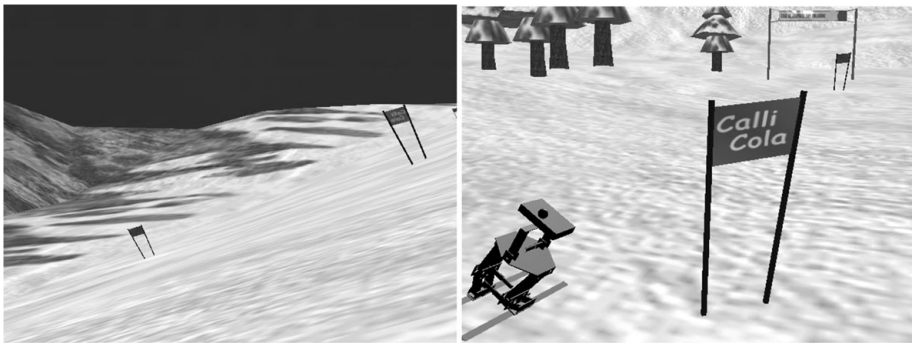


Fig. 11. Robot camera view and environment camera view of the same time instance.

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press

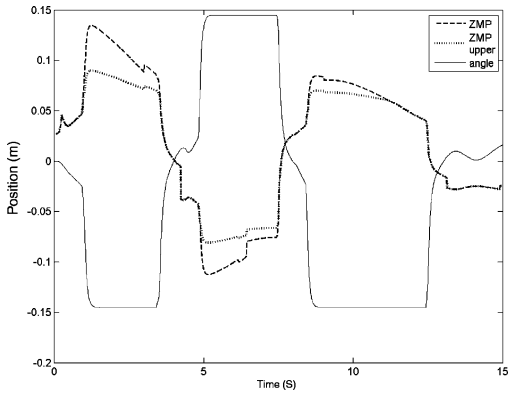


Fig. 14. Comparison of ZMPs with controlled and fixed torso.

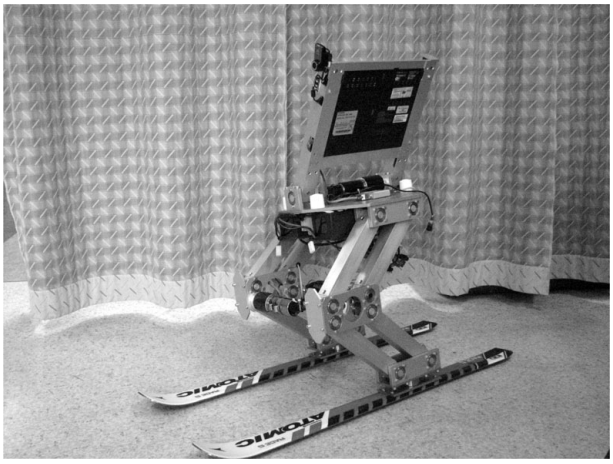


Fig. 15. Robot skier.

frequency of 1 kHz. Accordingly, we had to develop appropriate device drivers for the motion controllers and the electronic gyroscope.[12] The main computer communicated with the control computer using Ethernet. The task of the main computer was the path planning using the video camera, GPS sensors, and robot posture, provided by the control computer. Vision processing was accomplished using a single camera with a view angle of 78°. With an image size of 320×200 pixels we were able to obtain the object recognition and localization at 30 fps. The distance between the current robot position and the next gate was calculated using the camera orientation and the estimated size of the gate mark. We used the blue and red gate fences respectively. The color defined whether the robot-skier should approach them from the left or from the right side dependent on the marked color. According to the localization information the angular acceleration and consecutively also the desired turn radii was estimated. The velocity in the sagittal plane was obtained by the GPS sensor with accuracy of 0.1 m/s and sample time of 0.25 s.

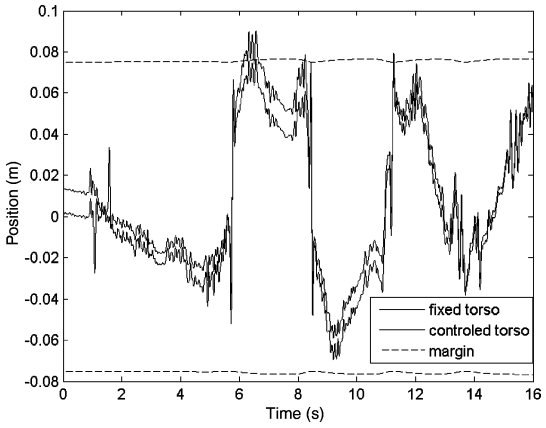


Fig. 17. y_{ZMP} deviation from y_{MSP}.

6.2. *Experiments*

We performed several experimental runs on the ski slope. Our experiments showed that the robot was able to follow the desired path marked by any combination of four race

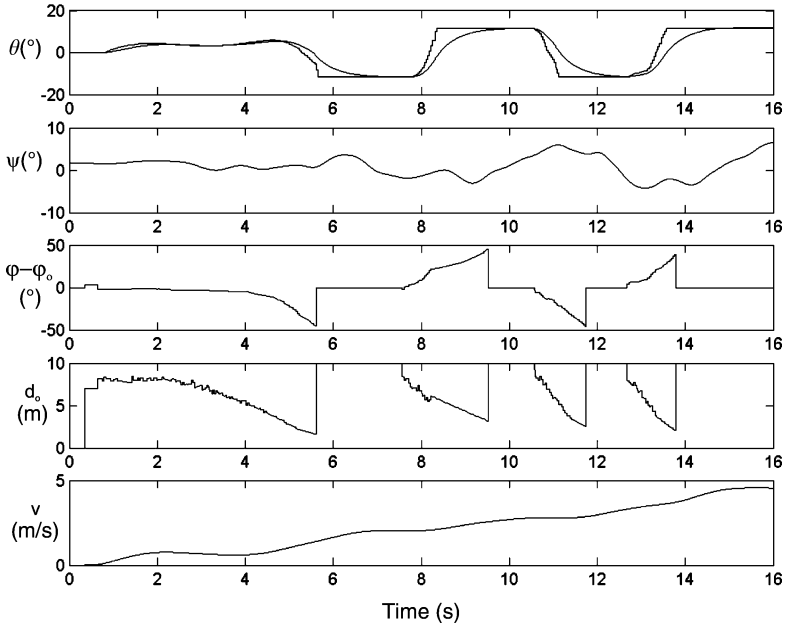


Fig. 16. Desired and real inclination (θ), local ground inclination (ψ), target angle ($\varphi - \varphi_o$), distance to the gate (d_o) and velocity in the sagital plane (v) during the test run.

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press

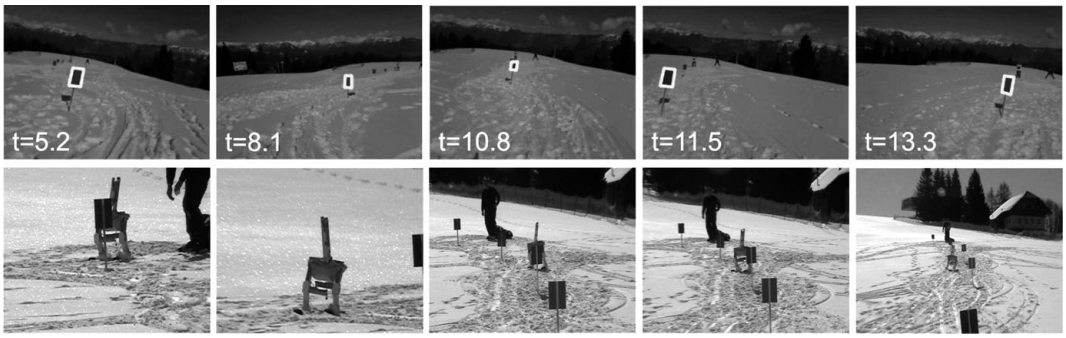


Fig. 18. Snapshots of robot and slope view.

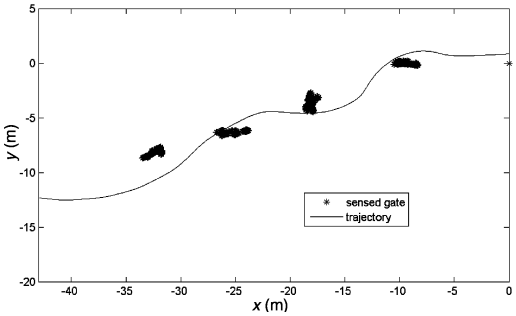


Fig. 19. Reconstructed path between gates.

gates providing that the distance between the gates allowed the robot to execute the turn.

Results from one typical test-run are shown in Fig. 16. In the upper plot we can see how the desired robot inclination (θ) varies with the increasing speed and how the stability control limited the robot inclination in order to assure stability. The next plot shows the measured local inclination (ψ) of the ski slope in the lateral plane. The next two plots show the target angle ($\varphi - \varphi_o$) and the distance to the next gate d_o, estimated from the vision system. We can see that the angle drops to 0 value and the distance is set to the maximal distance when the vision system looses the tracked object. During the experiments the robot reached maximal velocity of approx. 5 m/s, as can be seen in the bottom plot.

Figure 17 shows calculated position deviation of y_{ZMP} from y_{MSP} in the lateral direction. The experiments were reported also with the controlled torso. In order to show how the torso movement contributes to the stability, we also $calculated y_{ZMP}$ regarding the fixed torso position. At the start when the robot is not moving, y_{ZMP} for the fixed torso is not in the most stable position because there is local ground inclination (as can be seen from Fig. 16), while for the controlled torso y_{ZMP} is close to the y_{MSP}. One can also note that also during the skiing y_{ZMP} never exceeds the stability margin $0.5\, b_{sr}$ (doted line). Figure 18 shows the sequence of images captured from the robot and the fixed camera at the same time instances.

Finally, Fig. 19 shows the robot path obtained by the velocity integration. Red dots represent the estimated gate positions. Since neither the velocity neither the estimated gate positions were reliable, we obtained a group of points instead of a single point for the gate position.

7. Conclusions

In this paper, we presented a robot-skier capable of autonomous navigation on an unknown ski slope. One of the major challenges of such a system is the stability. We proposed and implemented an algorithm that provides the stability of the robot skier on an inclined surface based on ZMP. Our approach differs from the conventional ZMP based algorithms since we did not control ZMP exactly. Furthermore, we assured that the ZMP stays inside the support region. We proposed hierarchial control law, where the skier stability is of primary importance. The ski turns are performed in such a way, that skier stability is maximized. Additionally, the robot is compliant to the sudden shocks due to the terrain irregularities. Best results were obtained using the predicted ground reaction forces. In navigation, we enhanced the original human acceleration model by the implementation of virtual goals. It was demonstrated, that with this modification we obtain similar path as an experienced human skier would do. At the same time smooth transitions between gates are generated.

References

1. J. Baltes and N. Hildreth, "Adaptive Path Planner for Highly Dynamic Environments," **In:** *RoboCup 2000: Robot Soccer World Cup IV* (Melbourne, Australia, 2001) pp. 76–85.
2. J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE J. Robot. Automation* **7**(3), 278–288 (1991).
3. B. R. Fajen, W. H. Warren, S. Termizer and L. P. Kaebling, "A dynamical model of steering, obstacle avoidance, and route selection," *Int. J. Comput. Vis.* **54**(1–2), 13–34 (2003).
4. P. A. Federolf, Finite Element Simulation of a Carving Snow Ski *Ph.D. Thesis* (2005). At Swiss Federal Institute of Technology Zurich.
5. S. S. Ge, X. C. Lai and A. Al Mamun, "Sensor-based path planning for non-holonomic mobile robots subject to dynamic constraints," *Robot. Autonom. Syst.* **55**, 513–526 (2007).
6. K. Hasegawa, S. Shimizu and M. Yoshizawa, "Robotics applied to sports engineering," *Adv. Robot.* **14**(5), 377–379 (2000).
7. J. Howe, *The New Skiing Mechanics* (Mclntire Publishing, Waterford, ME, USA, 2001).
8. W. H. Huang, B. R. Fajen, J. R. Fink and W. H. Warren, "Visual navigation and obstacle avoidance using a steering potential function," *Robot. Autonom. Syst.* **54**(4), 288–299 (2006).
9. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.* **5**(l), 90–98 (1986).

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press

10. L. Lahajnar, A. Kos and B. Nemec, "Modelling and Control of Autonomous Skiing Robot," In *6th EUROSIM Congres on Modelling and Simulation*, Ljubljana, Slovenia (2007).
11. D. Lind and S.P. Sanders, *The Physics of skiing - Second Edition* (Springer, New York, USA, 2004).
12. D. Omrcen, "Developing Matlab Simulink and xPC Target Real-time Control Environment for Humanoid Jumping Robot," In *16th International Workshop on Robotics in Alpe-Adria-Danube Region, RAAD 2007*, Ljubljana, Slovenia (2007) pp. 18–23.
13. M. Takahashi and T. Yoneyama, "Basic ski theory and acceleration during ski turn," *Sci. Skiing II* **14**(5), 307–321 (2001).
14. I. Ulrich and J. Borenstein, "Reliable Obstacle Avoidance for Fast Mobile Robots," In *IEEE International Conference on Robotics and Automation 2003*, Belgium (1998) pp. 1572–1577.
15. I. Ulrich and J. Borenstein, "Vfh *: Local Obstacle Avoidance with Lookahead Verification," In *IEEE International Conference on Robotics and Automation*, (2000). Vol. 11, San Francisco, CA, USA, pp. 2505–2511.
16. M. Vukobratovic and B. Borovac, "Zero-moment point-thirty five years of its life," *Int. J. Humanoid Robot.* **1**(1), 157–173 (2004).

https://doi.org/10.1017/S0263574708004955 Published online by Cambridge University Press