**The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA**

Control and Navigation of the Skiing Robot

Bojan Nemec and Leon Lahajnar

Abstract— **The paper describes the control and the navigation of a skiing robot that is capable of autonomous skiing on a ski slope using the carving skiing technique. Based on a complex sensory system it is capable of autonomous navigating between the race gates, avoiding obstacles and maintaining a stable position during skiing on an previously unknown ski slope. The robot was tested on different ski slopes with various race gates combination.**

I. INTRODUCTION

Research community is looking towards new field of application of humanoid robots. 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[5], [6]. There were only few previous attempts to develop a robotic skier [3]. In most 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 [7]. Our work focuses on these two previously ignored problems. We designed a special 3 degrees of freedom (DOF) robot dedicated for skiing using the carving technique, which uses a complex sensory and 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 [6]. In our previous work we have demonstrated that the robot is capable of autonomous skiing on an unknown ski slope. In this paper, we propose improved navigation and control algorithms. Algorithms were verified on the ski slope at the different snow surfaces.

II. SYSTEM OVERVIEW

Carving skis were introduced in the late 80's and became widely used in the late 90's[7]. The main difference between the carving and traditional skis is that carving skis exploits their shape to make turns and do not require skidding in order to perform a turn as do traditional skis. The essential issue is that the curvature radius of carving skis skis is defined by the geometry of the ski. Curvature radius can be described by

$$r = \frac{\frac{L^2\cos\theta}{4} + \frac{h^2}{\cos\theta}}{2h} \qquad (1)$$

where coefficients L and h describe the geometry of the ski as can be seen in Figure 1 and curvature cord length is approximated by the ski length L. A similar equation was presented by Howe [4], who also accounted the penetration depth of the ski for the snow. Although the penetration has profound effect to the curvature radius, it is hard to measure or to predict on a ski slope. 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 [2]. Humans 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 one DOF and enables motion in the lateral plane. The kinematics of the 3 DOF ski robot mechanism is presented in Figure 2. Robot joints are controlled by DC motors. The overall weight of the robot including batteries and skis is 19 kg. The size end the weight of the robot is comparable to the 8 year old child, which enabled the usage of standard child skis for the experiments. More details of the robot design can be found in [6]. The robot is controlled with a hierarchically build multi processor computer system. The upper level controller is used for the navigation, the vision processing and the decision making. Upper level controller communicates with a GPS receiver and an 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.

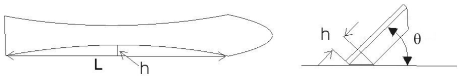

Fig. 1. Curvature radius of skis

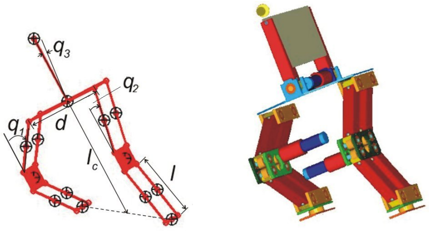

Fig. 2. Ski robot

III. CONTROL

The task of the robot skier control system is to assure tracking of a desired path while maintaining the stability

Robotics Laboratory, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
bojan.nemec@ijs.si

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

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 $\boldsymbol{q}_2$,

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

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

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

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

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

where

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

and

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

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 to 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:

$$\begin{bmatrix} \dot{r} \\ \dot{l}_c \end{bmatrix} = \mathbf{J}\dot{q}\ , \quad \mathbf{J} = \begin{bmatrix} \frac{a-b}{2h} & \frac{a+b}{2h} \\ -ls_1 & -ls_2 \end{bmatrix}\ , \tag{7}$$

where s_i denotes $\sin(\boldsymbol{q}_i)$. 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 used to stabilize the robot during the skiing.

A. 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 acting on an narrower support plane. Therefore we focused only on lateral stability. In accordance with the previously mentioned assumptions, we have used the simplified model of the skier in the lateral plane, as shown in Figure 3. The edging angle θ, which controls the curvature radius 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) [8] is in the middle between the skis. We will denote this position as MSP (most stable point). When the ZMP is outside of the support polygon, one ski loses the contact with the ground and the skier can fall. Therefore it is necessary to calculate ZMP of double inverted pendulum in inclined sufrace. 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 p=[$x_P \quad y_P$ 0]^T on the plane z=0 is

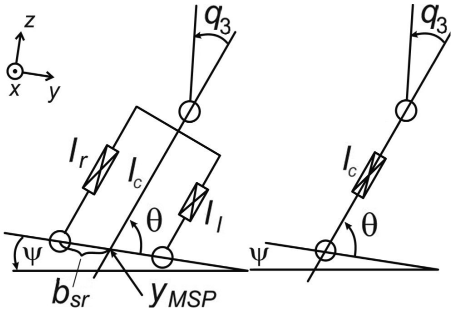

Fig. 3. Simplified model in lateral plane

$$\begin{aligned} &\sum_i (r_i-p)\times m_i(\ddot{r}_i+g)+\sum_i\left[I_i\dot{\omega}_i+\omega_i\times I_i\omega_i\right] \\ &-\sum_j M_j-\sum_k (s_k-p)\times F_k = M_P. \end{aligned} \tag{8}$$

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 (8) can be rewritten as

$$\begin{aligned} &\sum_i (r_i-p)\times m_i(\ddot{r}_i+g+a_{ri}) \\ &+\sum_i\left[I_i\dot{\omega}_i+\omega_i\times I_i\omega_i\right] = M_P, \end{aligned} \tag{9}$$

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

$$\sum_i (r_i-p)\times m_i(\ddot{r}_i+g+a_{ri})+T_i = M_P. \tag{10}$$

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

$$x_{ZMP} = \frac{\sum_i m_i\left(\ddot{z}_i+g_z\right)x_i - \sum_i m_i\left(\ddot{x}_i+g_x\right)z_i}{\sum_i m_i\left(\ddot{z}_i+g_z\right)} - \frac{-\sum_i\left(T_y\right)_i}{\sum_i m_i\left(\ddot{z}_i+g_z\right)} \tag{11}$$

and

$$y_{ZMP} = \frac{\sum_i m_i\left(\ddot{z}_i+g_z\right)y_i - \sum_i m_i\left(\ddot{y}_i+g_y+\frac{v_i^2}{R_i}\right)z_i}{\sum_i m_i\left(\ddot{z}_i+g_z\right)} - \frac{-\sum_i\left(T_x\right)_i}{\sum_i m_i\left(\ddot{z}_i+g_z\right)}. \tag{12}$$

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

$$M_P = M_{0,1} - p \times F_{0,1} \tag{13}$$

and the ZMP point is calculated by

$$x_{ZMP} = \frac{\left(M_{0,1}\right)_y}{\left(F_{0,1}\right)_z} y_{ZMP} = \frac{\left(M_{0,1}\right)_x}{\left(F_{0,1}\right)_z}. \tag{14}$$

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 objectives: 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 $\boldsymbol{q}_3$. When we can not compensate the excessive external forces with the torso, we have to use the legs. Consequently, we can not track the desired curvature radius. In this case the robot behavior is much like the human behavior - in order not to fall the skier changes the desired direction of the skiing. The control algorithm selects a proper action depending on the stability index and the current movement of the skier. The stability index is defined as

$$\Phi\left(\theta, \boldsymbol{q}_3, \psi, v\right) = 1 - \left(\frac{y_{MSP} - y_{ZMP}}{b_{sr}\left(\theta\right)}\right)^2, \tag{15}$$

where b_{sr} is the margin of stable region in the lateral direction and y_{MSP} is the most stable point as presented in Figure 3. In order to assure the stability also in presence of sudden and unexpected ground reaction shocks, we narrowed the available support polygon. The calculated ZMP is restricted to stay within the region

$$-\frac{d}{2cos(\theta)} + y_{MSP} \leq y_{ZMP} \leq \frac{d}{2cos(\theta)} + y_{MSP}. \tag{16}$$

In accordance with (15) the stability index should not fall bellow the value 0.75 in the case when we allow maximal $0.5b_{sr}$ deviation of the ZMP from the MSP. The overall control scheme is presented in Figure 4. 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 can not 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.

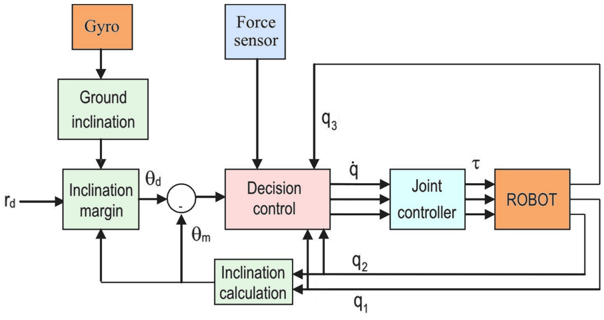

Fig. 4. Overall low level control scheme of skiing robot

The decision control block 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 can not 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

$$F_{e1} = \frac{F}{2}\left(1 - \frac{y_{ZMP}}{b_{sr}}\right), F_{e2} = \frac{F}{2}\left(1 + \frac{y_{ZMP}}{b_{sr}}\right). \tag{17}$$

The compliant behavior is obtained using the following control law

$$\dot{q}_i == \begin{cases} \mathbf{K}_c(F_{ei} - F_i + \sigma), & (F_{ei} - F_i) < -\sigma \\ 0, & -\sigma \leq (F_{ei} - F_i) \leq \sigma \\ \mathbf{K}_c(F_{ei} - F_i - \sigma), & (F_{ei} - F_i) > \sigma \end{cases} \tag{18}$$

where F_i, $i = \{1,2\}$ 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. However, it turned out that the active compliance is too slow to damp all shocks on the ski slope. Note that even humans can not effectively damp terrain irregularities without the vision feedback. Therefore, we have used also passive - mechanical compliance between the skis and the robot legs.

IV. NAVIGATION

Navigation is one of the most challenging areas in the real-time mobile robot control, especially for non-holonomic mobile robots. All previously proposed navigation methods for mobile robots assume that we can control the path velocity using the drive actuators. In contrast, in the case of the skiing robot, we can not explicitly control the velocity. Therefore, the path planning algorithm must adapt to the current velocity.

The skier robot is a typical non-holonomic system and the equation of motion can be described as

$$\dot{X} = v\cos(\varphi), \quad \dot{Y} = v\sin(\varphi). \tag{19}$$

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

$$\varphi(t) = \int \frac{v(t)}{r(t)} dt, \tag{20}$$

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 can not yet carve a turn into the snow. Therefore, for small edging angles ($\theta < 5^o$),the turning model is described with a simple relation $r = \theta^{-1}$.

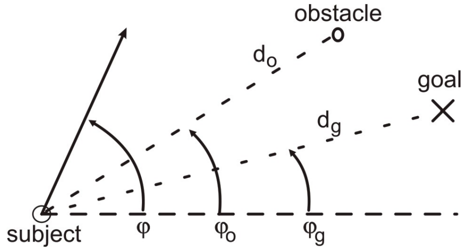

Fig. 5. Orientation of the subject in dependence of the goal and an obstacle

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 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 [1]. 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

$$\begin{aligned}\ddot{\varphi} = & -b\dot{\varphi} - k_g(\varphi - \varphi_c)\left(e^{-c_1 d_g} + c_2\right) \\ & + \sum_i k_{oi}(\varphi - \varphi_{oi})\left(e^{-c_3|\varphi - \varphi_{oi}|}\right)e^{-c_4 d_{oi}},\end{aligned} \tag{21}$$

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-the obstacle respectively, while $b, k_g, k_{oi}, c_1, c_2, c_3$ and c_4 are the tuned parameters (see Figure 5). (21) has three components.

- $-b\dot{\varphi}$ damping component
- $k_g(\varphi - \varphi_c)\left(e^{-c_1 d_g} + c_2\right)$ 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.
- $+\sum_i k_{oi}(\varphi - \varphi_{oi})\left(e^{-c_3|\varphi - \varphi_{oi}|}\right)e^{-c_4 d_{oi}}$ component for obstacle avoidance, that rebounds the subject from nearby obstacles

Our on-line path planing procedure is illustrated in Figure 6, 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 have to estimate also 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 have to estimate the distance between the robot and the next gate and the angle between the current robot direction and the next gate. For the estimation we use single camera mounted on the robot torso It is assumed that the left and the right gates are marked with the blue and red fence of equal size, respectively. The distance d to the next race gate is estimated through the size

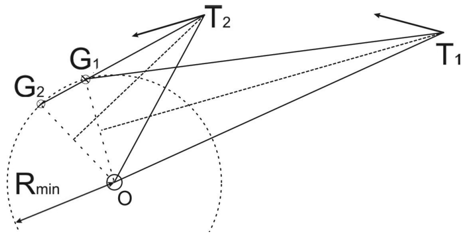

Fig. 6. Local navigation on the ski slope

of the fence. We have used edge detection technique, which detects all square objects rotated for angle $\theta + q_3 + \psi$ 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. From the image we estimate also the angle ϕ between the direction of the movement and race gated, which is proportional to the displacement of the race gates from the center of the captured image. Unfortunately, odometry using using camera fails whenever the gate fence is not visible, which is very often situation during the skiing. To overcome this problem we have applied Extended Kalman filtering, which optimally estimates states and output of an nonlinear system.

Let describe the system state with $p = f(X, Y, \varphi) + w$ and the system output with $z = h(d, \phi) + u$, where w and u are the state and output noise respectively. Robot position in the time instance t can be predicted from the position and velocities in the time instance $t-1$ using equation

$$\hat{p}_t = f(p_{t-1}, v_{t-1}, wt-1) \begin{bmatrix} X_t \\ Y_t \\ \varphi_t \end{bmatrix} = \begin{bmatrix} X_{t-1} \\ Y_{t-1} \\ \varphi_{t-1} \end{bmatrix} +$$

$$\begin{bmatrix} (v_{t-1,X} + w_{t-1,X}) cos\varphi_{t-1} - (v_{t-1,Y} + w_{t-1,Y}) sin\varphi_{t-1} \\ (v_{t-1,X} + w_{t-1,X}) sin\varphi_{t-1} - (v_{t-1,Y} + w_{t-1,Y}) cos\varphi_{t-1} \\ v_{t-1,\varphi} + w_{t-1,\varphi}\varphi_{t-1} \end{bmatrix} \quad (22)$$

where v are the robot velocities. Second subscript denotes the coordinate. At this instance we calculate also predicted error covariance matrix

$$\hat{P}_t = A_t P_{t-1} A_t^T + W_t Q_{t-1} W_t^T \ , \quad (23)$$

where $A = \frac{\delta f}{\delta p}$, $W = \frac{\delta f}{\delta w}$ and Q is estimated noise matrix. We have evaluated values of this matrix experimentally. Best results were obtained with Q as diagonal matrix with values $(0.2\Delta t)^2$, $(0.2\Delta t)^2$, $(0.02)^2$. Covariance matrix is initialized with values, that describe maximal expected errors for X, Y and φ respectively. From the predicted values we calculate extended Kalman filter updates using the equation

$$p_t = \hat{p}_t + K_t(z_t - h(\hat{p}_t)) \ , \quad (24)$$

where the output prediction is calculated with

$$h(\hat{p}_t) = \begin{bmatrix} \hat{d}_t \\ \hat{\phi}_t \end{bmatrix} = \begin{bmatrix} \sqrt{\hat{p}_{t,X}^2 + \hat{p}_{t,Y}^2} \\ atan(\frac{-\hat{p}_{t,X}}{\hat{p}_{t,Y}}) - \hat{p}_{t,\varphi} \end{bmatrix} ,$$

and the Kalman gain is

$$K_t = \hat{P}_t H_t^T S_t^{-1} \ , \ S_t = H_t \hat{P}_t H_t^T + R_t \ , \quad (25)$$

and

$$H_t = \frac{\delta h}{\delta p} \ , \ P_t = (I - K_t H_t)\hat{P}_t \ . \quad (26)$$

R_t is the camera noise. Kalman filter is initialized each time the robot passes the race gates and initial distance is set to 10m. Whenever the camera is not able to detect the next race gates, we use the speed information obtained from the GPS.

V. EXPERIMENTAL RESULTS

Experiments were performed on various ski slopes with the average inclination of 7 degrees, where also some terrain irregularities were present. Our experiments showed that the robot was able to follow the desired path marked by any combination of four race gates providing that the distance between the gates allowed the robot to execute the turn.

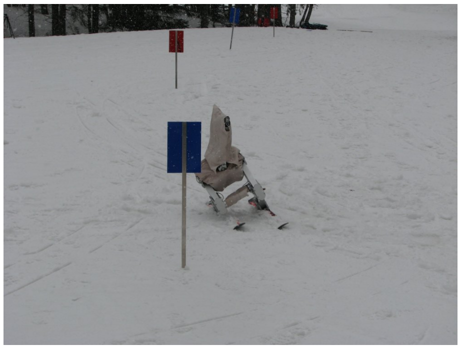

Fig. 7. Robot skier

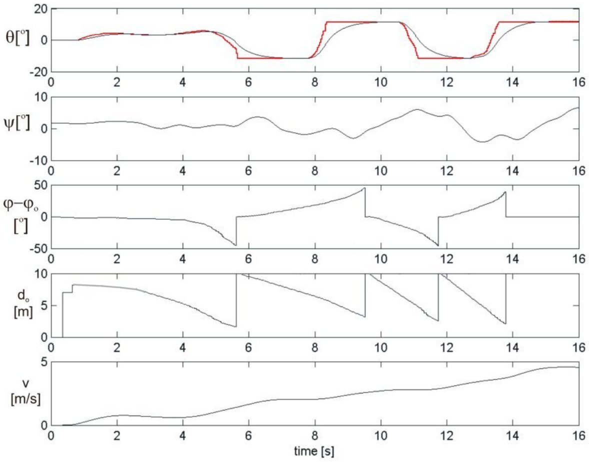

Fig. 8. Desired (red line) and real (black line) inclination (θ), local ground inclination (ψ), target angle ($\varphi - \varphi_o$), distance to the gate (d_o) and velocity in the sagital plane (v) during the test run

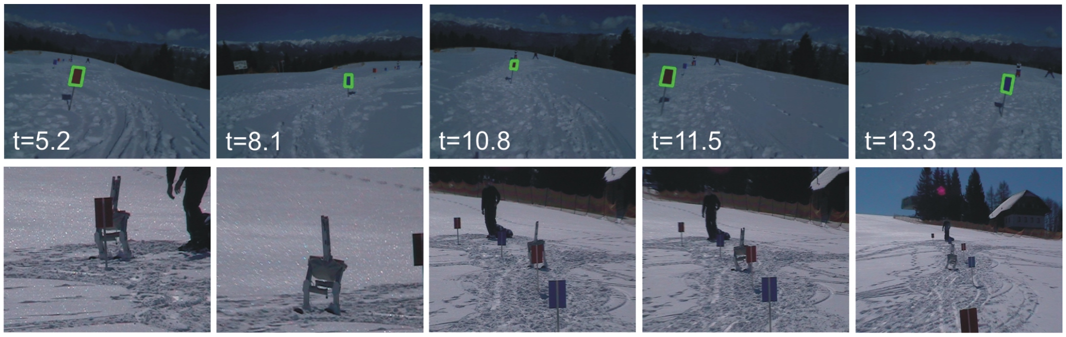

Fig. 10. Snapshots of the robot and slope view

Results from one typical test run are shown in Figure 8. In the upper plot we can see how the desired robot inclination (θ) varies with the increasing speed (red line) and how the stability control limited the robot inclination in order to assure stability (black line). 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 and Kalman filter. We can see that we can estimate these two values also 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 9 shows calculated position deviation of y_{ZMP} from y_{MSP} in the lateral direction. In order to show how the torso movement contributes to the stability, we calculated also y_{ZMP} regarding the fixed torso position. One can note that also during skiing y_{ZMP} never exceeds the stability margin $0.5b_{sr}$ (doted line). Figure 10 shows the sequence of images captured from the robot and the fixed camera at the same time instances. Finally, Figure 11 shows the robot path obtained by the velocity integration. Blue 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.

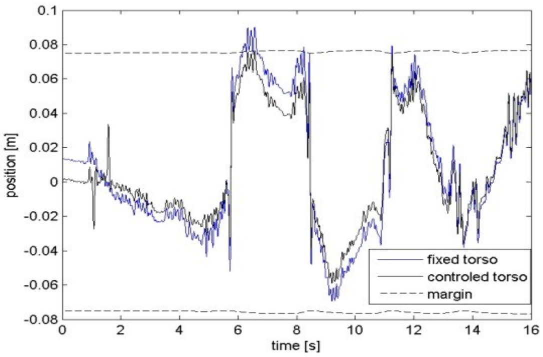

Fig. 9. y_{ZMP} deviation from y_{MSP}

VI. 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 preserved. Additionally, the robot is compliant to the sudden shocks due to the terrain irregularities. 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.

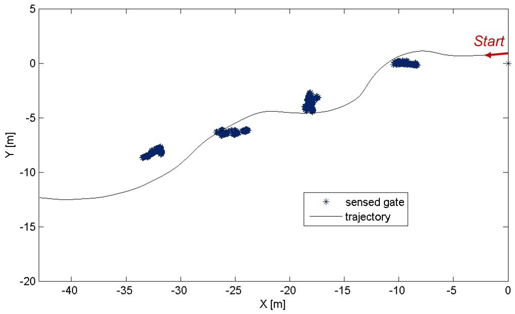

Fig. 11. Reconstructed path between the gates

REFERENCES

[1] B. R. Fajen, W. H.Warren, S. Termizer, and L. P. Kaebling. A dynamical model of steering, obstacle avoidance, and route selection. *International Journal of Computer Vision*, 54(1-2):13–34, 2003.

[2] P. A. Federolf. *Finite element simulation of a carving snow ski*. Phd. Thesis, 2005.

[3] K. Hasegawa, S. Shimizu, and M. Yoshizawa. Robotics applied to sports engineering. *Advanced Robotics*, 14(5):377–379, 2000.

[4] J. Howe. *The New Skiing Mechanics*. McIntire Publishing, 2001.

[5] 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.

[6] L. Lahajnar, A. Kos, and B. Nemec. Skiing robot - design, control, and navigation in unstructured enviroment. volume [In Press], 2009.

[7] D. Lind and S.P. Sanders. *The Physics of skiing - Second Edition*. Springer, 2004.

[8] M. Vukobratovič and B. Borovac. Zero-moment point-thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.