Robotica (2014) volume 32, pp. 413–431. © Cambridge University Press 2013
doi:10.1017/S0263574713000684

A novel approach for humanoid push recovery using stereopsis

Mohammad-Ali Nikouei Mahani†,‡, Shahram Jafari†,* and Hadi Rahmatkhah†

†*School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran*
‡*School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran*

(Accepted June 17, 2013. First published online: August 7, 2013)

SUMMARY
Push recovery is one of the most challenging problems for the current humanoid robots. The importance of push recovery can be well observed in the real environment. The critical issue for a humanoid is to maintain and recover its balance against any disturbances. In this research a new stereovision approach is proposed to estimate the robot deviation angle and consequently, the movement of center of mass of the robot is calculated. Then, two novel strategies have been devised to recover the balance of the humanoid which are called "knee strategy" and "knee-hip strategy." Also, a mathematical model validates the efficiency of the proposed strategies as demonstrated in the paper. Experiments have been conducted on a humanoid robot and demonstrate that the predicted robot deviation angle, using stereovision technique, converges to the actual deviation angle. Stable regions of proposed strategies illustrate that the humanoid can recover its stability in a robust manner. Vision-based estimation also shows a higher correlation to actual deviation angle and a lower fluctuation compared with the output of the acceleration sensor.

KEYWORDS: Humanoid robot; Push recovery; Stereovision technique; Optical flow; Deviation angle; Knee strategy; Knee-hip strategy.

1. Introduction

In the past recent years, humanoid robot is known as the state of the art in advanced robotic systems.[1,2] Designing and controlling of humanoid robots become a challenge and many new research topics were defined on humanoid robots. Among these new topics, stability control and push recovery are known as significant topics in real environment.

Push recovery is the task of detecting, analyzing, and finding a proper strategy to recover the balance of the robot against any push.[3] The push force may have arbitrary reasons. For instance, it may come from a collision, walking on rough terrain, or any other things. Without caring about what the source of the push is, the main challenge is to estimate the direction and the magnitude of the push and to find a proper strategy to recover the balance of the robot.

Unseen real environment parameters and situations contribute to the importance of push recovery in humanoid robots. Humanoid robots may contact or collide with other humanoids, obstacles, and any other things. They may have to walk on a rough terrain in the real environment. Consequently, they should recover their stability in such situations in order to perform their mission successfully. Push recovery becomes far more important when the missions of humanoid robots are critical.

Robotic soccer competitions are the other examples which illustrate the importance of push recovery. Since the robots should perform as quickly as possible during competitions they may lose the time or a proper rank in case they fall because of any reason.

Prevalent sensors which are used to estimate the dynamic state of a humanoid robot can be categorized in three groups: acceleration and tilt sensors, joint torque/angle-reader sensors, and

* Corresponding author. E-mail: jafaris@shirazu.ac.ir

feet-force sensors. Usually, acceleration sensors are installed in the hip to estimate the acceleration of the center of mass (COM) of a humanoid robot. Joint torque/angle-reader sensors can be merged to motor control systems, e.g., new digital servo motors, or installed as distinct sensors on the joints. Feet-force sensors are the other type of sensors which are installed under the feet of the robot.[4,5] Feet-force sensors can help to evaluate the position of the COM by estimating the ground reaction force or estimating the center of the pressure (COP). Feet-force sensors are not adapted for walking on a rough terrain since the sensors may not completely contact with the floor. Usually, acceleration sensors' outputs are fluctuating and inaccurate.

In this paper, a new category of the sensors which can be used in humanoid robot push recovery is presented. Indeed, this new category is a vision-based method, which is used to estimate the dynamic state of a humanoid robot. The proposed method uses stereovision technique and will be explained in details later. Albeit, camera is not a new sensor, using it in estimating the dynamic state of a robot for push recovery is a new point of view.

The proposed vision-based method can be used in conjunction with the other sensors or can be used standalone as a separate method.

Stereo vision and 3D vision is used in many researches for humanoid robots. Kagami *et al.*[6] used 3D vision in their H7 humanoid robots to interact with human being, obstacle avoidance, and target grasping. Vision is not used to recover the stability in their research.

Stereo vision is used by Kanzaki *et al.*[7] to predict the clash timing of a moving object. Their method estimates the distance of the moving object to the robot to estimate the clash time and use it in balance control. Although they used stereo vision technique, they did not use it for estimating the dynamic state of the robot and it was only used to predict the clash timing of a moving object which comes toward the robot.

Davison *et al.* proposed a method for simultaneous localization and mapping (SLAM) with a single camera. Their method runs at 30 Hz and their vision method used in HRP-2 humanoid robot.[8] Indeed they used an additional wide-angle camera and used it only for SLAM.

The next step in push recovery, after estimating the dynamic state of the robot, is analyzing and finding a proper strategy to recover the balance of the robot. A specific dynamic model should be assumed for the robot to analyze the dynamics of the robot based on it.

Exact modeling and analysis of a humanoid robot are very complex and expensive due to its complex dynamic behavior, high degree of freedom, and natural instability. In most research, a simplified model of the robot is considered for stability analysis and control.[9,10] This simplification will disregard some useful information that can help the robot to recover the stability or act more smoothly. Legs are the most important part of a humanoid robot for stability analysis, so simplification in legs may cause the loss of useful information. In most push recovery research, humanoid robots are considered as a linear inverted pendulum model (LIPM)[9–11] and robot stability analysis is carried out based on this assumption. LIPM makes stability estimation and designing of a controller to recover the stability much easier.

Yamamoto *et al.*[12] also assumed the LIPM as the robot model and applied the switching control framework to a center of gravity-zero moment point (COG-ZMP) to improve the robustness. He did not estimate the COM state in his work and used COG-ZMP.

Hofmann[13] proposed three models for recovering the stability in his thesis based on motions of the COM. He highlighted that the motion of COM in the horizontal direction is important.

Velocity formulations were extended by Pratt for biped walking in fast motions.[14] He also used both LIPM and flywheel model to propose a method to find the capture point for push recovery.[15] Push recovery problem was analyzed by Stephens *et al.* as well.[1] He illustrated three strategies for push recovery and defined a region for each strategy. Stephens used the regions to apply a proper strategy according to the current dynamic state of the robot to recover the stability. In the first region, which is known as the stable region, he showed that by assuming the robot as a LIPM, the robot will recover the stability by itself and no control is needed to recover the stability. In the second region, ankle strategy is used to recover the stability by controlling the ankle joint. In the third place, the hip of the robot was modeled as a flywheel. By this new assumption, a new stable region was presented which the robot can move back to the stable state against larger pushes compare with the previous strategy. The last strategy was stepping and it is used when the robot is out of the other strategies' regions. Stepping strategy is used for the largest disturbances and robot stability is recovered by taking a step. Many methods are proposed to select a proper point in stepping method. Xia *et al.* proposed a new

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

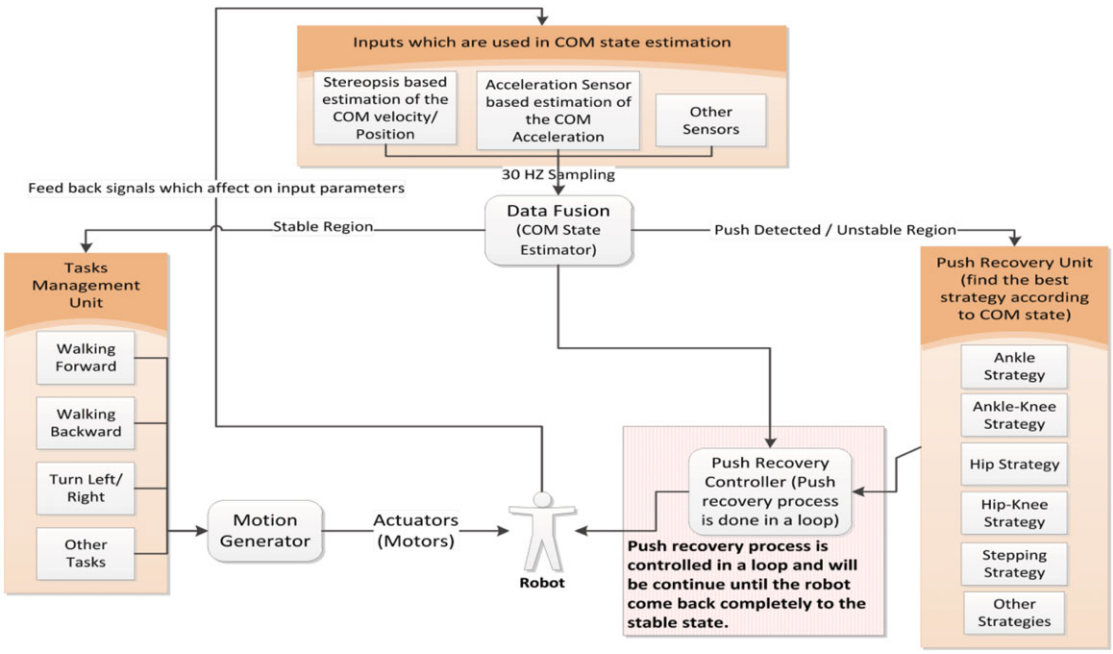


Fig. 1. (Colour online) The proposed framework for push recovery in a nutshell.

footstep planning method using sampling-based footstep planners.[16] Also, Rebula *et al.*[17] proposed a method for finding the proper capture points in a stepping push recovery. He believes since model assumptions are simple, modeling errors can cause to step in a wrong place. He proposed a method for learning the offsets of the real capture points to the predicted capture points which are predicted by LIPM assumption.

In this paper, in addition to a new stereovision-based technique for estimation of the COM state, two new strategies for push recovery which are called "knee" and "knee-hip" are presented. In these strategies, unlike LIPM, knee joint is considered to establish the model and robot uses this joint in addition to other joints to recover the stability. In the proposed strategies, knee is considered as an effective joint and it influences the recovery process. Experiments illustrate that knee and knee-hip strategies can improve the push recovery regions which are obtained by Stephens *et al.* By using knee and knee-hip strategies, robot can recover the stability against larger disturbances. Therefore, robot is not forced to take a step to recover the stability against larger pushes and it can use knee and knee-hip strategies instead. Indeed, stepping is not a proper strategy for push recovery, since it will change the location of the robot and robot may change its mission to recover the stability. In this research two new strategies are proposed to recover the stability of the robot against larger pushes. The proposed strategies illustrate that knee can play a significant role in a stability recovering process. In this paper, we do not focus on stepping and propose the strategies that do not change the robot location and do not need stepping.

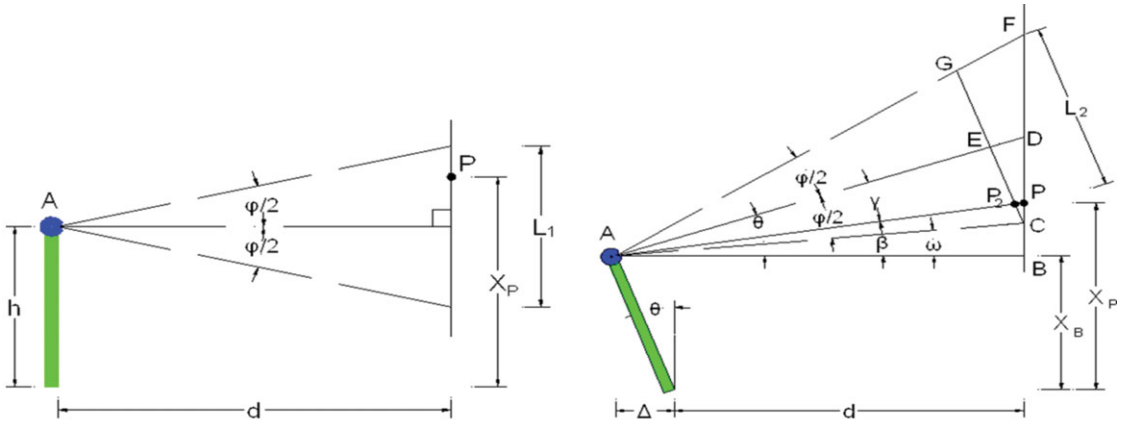
Figure 1 illustrates framework of the proposed push recovery in a nutshell. As it is obvious in Fig. 1, inputs of the framework are captured 30 times in a second. In each time, data fusion part estimates the state of the COM by fusing inputs.

The fusion algorithm which is used in the experiments is inspired by well-known robust averaging algorithm.[18]

Framework evaluates the stability of the robot by knowing the state of the COM. In case the robot is in the stable region, task management unit (TMU) will control the robot motions and behaviors. TMU sends the proper commands to the motion generator unit. Motion generator unit controls the actuators of the robot to carry out the TMU command.

In case the robot is out of the stable region, push recovery unit (PRU) will control the robot motions. In the first stage, PRU specifies the proper strategy for push recovery based on the COM state. Second, the push recovery controller unit guides the robot actuators to recover the stability based on the strategy which is specified by PRU.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press



(A). Humanoid robot in standard upright position. (B). Robot state after push disturbances.

Fig. 2. (Colour online) Deviation angle variation of a humanoid after a push. (A) illustrates a humanoid in standard upright position and (B) how the robot deviates after a push. The goal is to estimate the angle variation of the robot after a push.

TMU or PRU controls the robot actuators indirectly and variation of the robot actuators affects on inputs which are used in COM state estimation. The updated COM state is used to control the robot actuators again. This control loop is running at 30 Hz. In case the robot is out of the stable region, the push recovery controller unit will guide the robot until the stability is recovered completely. Then, the framework comes back to the normal loop and the stability of the robot is checked repeatedly.

The rest of the paper is organized as follow. In Section 2, a stereovision approach will be proposed for estimating the motions of the COM. Next topic discusses the knee and knee-hip strategies. These strategies are defined based on the motions of the COM and this section relies on the previous section to choose the best strategy for recovery. Indeed, Section 2 is the first stage of a push recovery system, which detects and analyzes a push, and Section 3 is the second stage, which recovers the robot against the push. Sections 2 and 3 describe a framework for push recovery. Section 4 is the experimental results and it illustrates that the estimated deviation angle using the vision approach converges to the deviation angle measured using the oscillating platform. Also, the proposed strategies are compared to each other and to other strategies in this section. Section 5 is the conclusion of the proposed push recovery framework.

2. Vision-Based Estimation of the Robot Deviation Angle

In this section, estimation of the deviation angle of the robot using the stereovision technique is studied in details.

Although much research was carried out to make the captured images more robust against vibrations,[19] our experiments did not suffer from this problem since the proposed vision algorithm runs at frame rate of the ordinary cameras (each 30 fps), cameras specification and installation are proper, and the robot size is not large. Head vibrations and blurred images are issues, which some humanoids faced with. However, these issues are not concerned in our experiments. A movie which illustrates the vision scene of our robot, when it walks or vibrates, is available at: http://www.mahani.info/Omid/RobustVision.

According to this movie, one can see that blurred images did not appear in the experiments, so paper ignores to discuss more on this issue.

This section is divided and studied in three subsections. In the first subsection, the simple method for estimating the deviation angle is presented. Second, the stereovision technique, which makes the algorithm applicable in real environment, is illustrated. Finally, a method is shown to convert the deviation angle variation into COM state.

2.1. Simple estimation of the deviation angle of the robot

To estimate the deviation angle of the robot, the position variation of a specific point, like P, is analyzed from the robot perspective. Figure 2(A) illustrates the humanoid robot in its standard balanced stand

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

state and (B) shows it after a push when it deviates θ degrees. As it can be seen in Fig. 2, robot is considered as a LIPM. The goal is to estimate θ, which shows the variation of the deviation angle, using the position variation of the point P from the robot perspective.

In Fig 2, the robot is simplified as a bar, A is the camera position and the height of the robot is h. d is the real distance between the robot and the point P. φ is the vertical view angle of the camera. Suppose the robot captured the image I_1 in standard upright position, and captured the image I_2 after a push, when it was deviated at angle θ.

In case the position of P in the image I_1 is x_P. After a push, P has a new position in the image I_2 which is denoted by x_{P2}. If the height of the image is l, (1) shows the relationship between X_P, the real position of P, and x_P, the position of P in the image.

$$x_P = \frac{X_P - h}{L_1} l. \tag{1}$$

Respectively, x_{P2} can be achieved by (2) as follows:

$$x_{P2} = \frac{EP2}{L_2} l = \frac{-l\,(\tan\theta - \tan\beta)}{2\tan\frac{\varphi}{2}\,(1 + \tan\theta\tan\beta)}. \tag{2}$$

According to Fig 2, $\tan\beta$ can be calculated by (3):

$$\tan\beta = \frac{2d\tan\frac{\varphi}{2}x_P + h\,(1-\cos\theta)\,l}{(d + h\sin\theta)\,l}, \tag{3}$$

where θ deviation angle can be estimated using (2) and (3) by two consequent images (I_1 and I_2) which are taken from the robot camera. The angular velocity of the robot would be determined using the time between these two consequent images.

To find the proper θ, (4) is written using (2) and (3) (see the Appendix for more details). A root for (4), is the proper θ, where all other parameters of (4) are available due to robot specifications.

$$\frac{-l\left(\tan\theta - \dfrac{2d\tan\frac{\varphi}{2}x_P + h\,(1-\cos\theta)\,l}{(d + h\sin\theta)\,l}\right)}{2\tan\frac{\varphi}{2}\left(1 + \tan\theta\dfrac{2d\tan\frac{\varphi}{2}x_P + h\,(1-\cos\theta)\,l}{(d + h\sin\theta)\,l}\right)} - x_{P'} = 0. \tag{4}$$

Indeed, (4) illustrates the velocity of the deviation angle since it shows the change of the deviation angle during a push. It illustrates the differences between the deviation angles before and after the push. Integral of the velocity of the deviation angle, gives the angle of the robot in a specific time. It can be seen in details in the experimental results section.

Corners are known as one of the best features for tracking purposes. In this way, Shi and Tomasi[20] proposed an improved version of Harris[21] corner detection, which is known as good features to track. We used Shi-Tomasi method to find the proper pixels for tracking.

To track the point P from image I_1 to image I_2, the sparse optical flow method is used. Optical flow is a technique to estimate the motions of the corners, objects, edges, and the like between two images.[22,23] Dense optical flow techniques do not perform adequately for all pixels. They may fail in same intense regions like a white wall in a room. In addition, dense optical flow methods have more time complexity in comparison with sparse optical flow techniques. For a regular use of optical flow, a fixed camera captures the images of the scene containing moving objects. However optical flow is utilized in this research with a different assumption: moving camera and fixed environment. Cameras are placed on the head of the robot and move according to the robot motions. Motions of the robot in the form of deviation angles can be estimated by the optical flow. The pyramid implementation[24] of the Lucas Kanade optical flow algorithm was used in our approach. It starts the tracking from highest level of the image, which has the lowest details, and working down to the lower level which has the highest details of the image. Pyramids cause to catch the large motions by using local windows.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

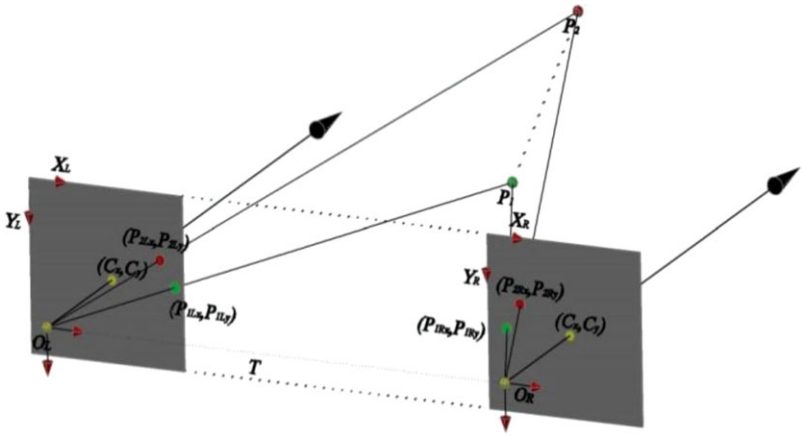


Fig. 3. (Colour online) The model of the cameras in stereovision approach to estimate the distance of a pixel.

2.2. *Stereovision*

The main necessity for using the stereovision technique is to estimate the real distance between the robot and the tracking pixels. All parameters of (4) are available according to the robot and the camera specifications except the real distance, d, between the robot and the pixel P. In case the proposed method is intended to be applicable in real environment, it should be independent from any environment parameters. The stereovision technique can estimate d and makes the proposed algorithm much more applicable in real environment and real experiments. The common stereovision techniques calculate the disparity image and the real distance can be estimated according to the disparity image and the geometric parameters of the cameras. Although calculating the disparity image is an adequate technique and gives much information about the environment, it is complex and not suitable for real time applications especially for our purpose. Consequently, a modified version of the stereovision technique is used which works on a number of specific pixels only. Back to the previous subsection according to Shi-Tomasi method, a number of pixels (tracking pixels) should be extracted for tracking, and before using optical flow. The distance of tracking pixels should be estimated to complete the requirements of the (4). Therefore, the stereovision technique should be modified to estimate only distances of the tracking pixels instead of finding the whole disparity image. Figure 3 illustrates the camera model in stereovision technique which is used to estimate the distance of a pixel like P.

In ordinary applications, cameras are fixed and the object moves.[25] Consequently, 3D tracking of an object is done based on fixed cameras assumption. However, the situation is vice versa. In our application the object is considered fixed and the camera moves. To find the distance of the tracking pixels, we used the pyramid optical flow method again. This time the flow is calculated between the right and the left image and the corresponding tracking points of the left image are found in the right image.

By knowing the camera geometric arrangement and using the triangulation, the depth map can be reprojected into the distance. Equation (5) shows the distance of point P_1 which is denoted as Z_{P_1}.

$$Z_{P_1} = \frac{fT}{P_{1Lx} - P_{1Rx}}, \tag{5}$$

where f is the focal length of the cameras, T is the distance between the principle points O_R and O_L. P_{1Lx} is the projection of P_1 on the left image plane and P_{1Rx} is the projection of P_1 on the right image plane. $(P_{1Lx} - P_{1Rx})$ is known as depth map (disparity value) and (5) converts it to the real distance value. The distance of P_2 can be estimated similar to P_1.

2.3. *Position/velocity estimation of the COM*

The stability condition of a robot is dependent on the initial position and velocity of the COM of the robot in many push recovery analysis.[1,4,5] COM motion estimation was used by Stephens[1] and he defined three regions for push recovery based on the initial position and velocity of the COM.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

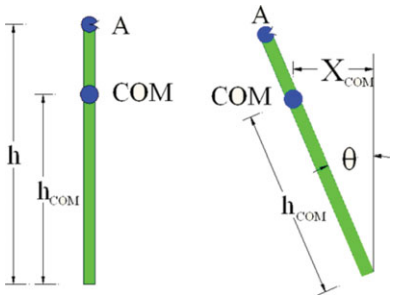


Fig. 4. (Colour online) Height of COM of the robot before and after disturbance.

According to these conditions one can select a suitable strategy to recover the stability of the robot. In the previous section the deviation angle of the robot was estimated by vision techniques.

Since in many researches the motions of the COM is the main key point to select the proper push recovery strategy, in this section the position and velocity of COM are computed from the deviation angle.

According to Fig. 4, the position of COM (X_{COM}) is determined by the following equation:

$$X_{\mathrm{COM}} = h_{\mathrm{COM}} \sin\theta. \tag{6}$$

Velocity of COM can be calculated from similar formulations:

$$\dot{X}_{\mathrm{COM}} = h_{\mathrm{COM}}\dot{\theta} \cos\,\theta. \tag{7}$$

In (7) $\dot{\theta}$ is unknown, one way to solve this difficulty is to calculate velocity from the position of COM by using the numerical relation:

$$\dot{X}_{\mathrm{COM}} = \frac{X_{\mathrm{COM}}(t_2) - X_{\mathrm{COM}}(t_1)}{t_2 - t_1}. \tag{8}$$

When the position and velocity of COM is determined, the stability region of the robot can be estimated.[1,5]

3. Push Recovery Strategies

Push recovery strategies are the final step of returning a robot to the balance state. Much research was carried out in this scope and several strategies were proposed and developed.

One of the main challenges is to choose a proper strategy according to the push strength and current robot state. Indeed, each strategy is effective in a specific situation. The effectiveness of the strategy is determined by a region which is defined according to the initial robot dynamic conditions. For instance, Stephens[1] proposed three strategies for push recovery and defined a stable region for each strategy based on the initial position and velocity of the COM of the robot. Stephens' proposed strategies are: (1) ankle strategy, (2) hip strategy, and (3) stepping.

Ankle strategy is effective for small pushes and recovers the robot stability using the ankle joint only. In this strategy, robot is considered as a LIPM and the COM of the robot is located at the hip. For bigger pushes, hip strategy is proposed and the robot recovers its stability using both hip and ankle joints. In this situation COM is modeled as a flywheel. For huge pushes which the hip strategy is not capable, robot recovers its stability using stepping strategy. In stepping strategy robot takes a step to avoid falling and the robot is forced to change its position. Consequently, stepping strategy is not an ideal strategy and is used only when no other strategy can be applied.

In this section, we ignore the LIPM assumption to suggest and study two new strategies for push recovery. In the proposed strategies, knee is considered as an effective joint and influences in the recovery process. The proposed strategies are called: "knee strategy" and "knee-hip strategy". In this section, these strategies are studied in details.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

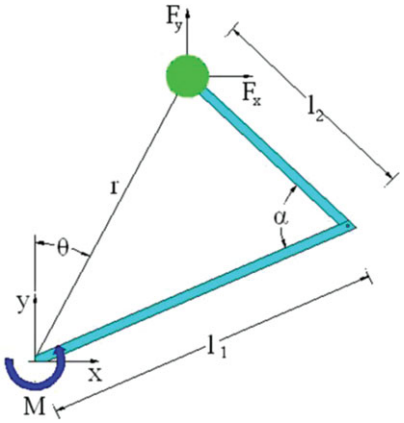


Fig. 5. (Colour online) Free body diagram of the robot in knee strategy. In free body diagram the external forces and reactions are shown, so the equilibrium equation is written easily.

3.1. Knee strategy

In knee strategy, LIPM assumption is broken. So knee joint is considered a variable in dynamic and kinematic models of the robot. Figure 5 illustrates the model of the robot in knee strategy which is called "knee model".

In the knee model, m is the mass of the robot, α is the knee joint angle, θ is the robot deviation angle, and the coordinate center is considered at the ankle of the robot.

According to Fig. 5, the geometric relation of the robot is written as follow:

$$r^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos\alpha. \tag{9}$$

First and second derivative of the r with respect to the time are illustrated in (11), and (12) as follows:

$$2r\dot{r} = 2l_1 l_2 \dot{\alpha} \sin\alpha \tag{10}$$

$$\dot{r} = \frac{l_1 l_2}{r}\dot{\alpha}\sin\alpha \tag{11}$$

$$\ddot{r} = \frac{l_1 l_2 \ddot{\alpha}\sin\alpha + l_1 l_2 \dot{\alpha}^2 \cos\alpha - \dot{r}^2}{r}. \tag{12}$$

Until now, the equations of r are written according to the scalar value of the r only. To write the dynamic equation of the robot, the polar coordinate system is used. So, the vector of r in the polar system is considered as:

$$\vec{\mathrm{r}} = \mathrm{r}\hat{e}_r, \tag{13}$$

where $\hat{e}_r$ is the unit vector in r direction. Consequently, the velocity and the acceleration of the COM are written as follow:

$$\frac{\mathrm{d}\vec{\mathrm{r}}}{\mathrm{d}t} = \dot{r}\hat{e}_r + r\dot{\theta}\hat{e}_\theta$$

$$\frac{\mathrm{d}^2\vec{\mathrm{r}}}{\mathrm{d}t^2} = \ddot{r}\hat{e}_r + \dot{r}\dot{\theta}\hat{e}_\theta + \dot{r}\dot{\theta}\hat{e}_\theta + r\ddot{\theta}\hat{e}_\theta - r\dot{\theta}^2\hat{e}_r \tag{14}$$

$$\vec{a} = \overrightarrow{\dot{r}} = (\ddot{r} - r\dot{\theta}^2)\hat{e}_r + (2\dot{r}\dot{\theta} + r\ddot{\theta})\hat{e}_\theta. \tag{15}$$

The inertia force can be calculated based on the acceleration of the COM. Therefore, the torque caused by the inertia force, which is denoted as$\vec{\tau}_1$, is obtained as follow:

$$\vec{F} = -m\overrightarrow{\dot{r}} \tag{16}$$

$$\vec{\tau}_1 = \vec{r} \times \vec{F} = m(2r\dot{r}\dot{\theta} + r^2\ddot{\theta})\circlearrowleft \tag{17}$$

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

Similarly, the torque caused by the gravity, $\vec{\tau}_2$, is obtained as follow:

$$\vec{\tau}_2 = -mgr \sin\theta \circlearrowleft, \tag{18}$$

where g is the acceleration due to the gravity. According to the equilibrium equation in vertical direction, the ground reaction force (GRF) in the y direction, R_y, is obtained as follow:

$$R_y = mg + ma_r \cos\theta - ma_\theta \sin\theta \tag{19}$$

$$R_y = mg + m(\ddot{r} - r\dot{\theta}^2)\cos\theta - m(2\dot{r}\dot{\theta} + r\ddot{\theta})\sin\theta. \tag{20}$$

Consequently, the torque due to the GRF is denoted as $\vec{\tau}_3$ and is obtained as follow:

$$\vec{\tau}_3 = mg\delta + m(\ddot{r} - r\dot{\theta}^2)\cos\theta\delta - m(2\dot{r}\dot{\theta} + r\ddot{\theta})\sin\theta\delta\sigma \circlearrowleft, \tag{21}$$

where δ is the distance between the ankle joint (the coordinate center) and the location where the GRF is applied. The magnitude and line of action of the reaction force R_y play an important role in the equilibrium of the robot. The distance between line of action of R_y and the ankle joint is restricted by the dimensions of the robot feet, but as shown in (20) the magnitude of the reaction force can be increased. There are five terms in the right-hand side of (21), if r is increased the second term cause an increase in the reaction force and forth term causes a decrease in the reaction force. The overall change in the reaction force is dependent to the initial condition, for $\theta \ll 1$, $\cos\theta = 1$ and $\sin\theta = \theta$, the reaction force is increased and the equilibrium region can be extended.

Since the thickness of the sole of the robot is negligible, the torque due to the horizontal component of GRF is assumed zero. In the equilibrium state, since the direction of $\vec{\tau}_3$ is opposite to the direction of $\vec{\tau}_1$ and $\vec{\tau}_2$, (22) should be consistent. Consequently, the dynamic equation of the robot in the knee model is illustrated in (23) (see the appendix for more details on how to solve the equation).

$$\vec{\tau}_1 + \vec{\tau}_2 + \vec{\tau}_3 = 0 \tag{22}$$

$$m(2r\dot{r}\dot{\theta} + r^2\ddot{\theta}) - mgr\sin\theta = mg\delta + m(\ddot{r} - r\dot{\theta}^2)\cos\theta\delta - m(2\dot{r}\dot{\theta} + r\ddot{\theta})\sin\theta. \tag{23}$$

Based on the equations of the Section 2.3, the angle/velocity of the deviation of the robot can easily be converted to the position/velocity of the COM.

Figure 6 illustrates the stable region of the knee strategy. Stable region of a strategy is defined based on the initial position and velocity of the COM and it contains the initial conditions where robot can recover its stability. Indeed, in case of the robot initial conditions, position/velocity of the COM, are in the stable region of a strategy, robot can recover its stability using the strategy. Later in the experimental results section, knee strategy stable region is compared to ankle strategy which is proposed by Stephens. The balance of the robot is recovered using the artificial potential function. Artificial potential functions are used in robot control and robot navigation. Much research focused on potential functions[26–28] to control the robot, navigations, etc. The gradient descent method is used to minimize the defined artificial potential error function. Figure 7 illustrates the trajectories which are controlled by the mentioned method.

The artificial potential error function, which is used in knee strategy control, is defined as follows:

$$F = \left\| Z - \hat{Z} \right\|, \tag{24}$$

where $Z = [X_{\text{COM}} \quad \dot{X}_{\text{COM}}]^T$ represents the current state of the COM, and $\hat{Z} = [\hat{X}_{\text{COM}} \quad \hat{X}_{\text{COM}}]^T$ is the desired stable state for COM.

In order to use a gradient descent based approach to find the optimal solution of F(i.e.,$Z = \hat{Z}$), a numerical method is used to find the gradient direction of F.

In knee strategy, ankle and knee joints are controlled to recover the stability of the robot. In fact, r and δ are the parameters which are controlled to recover the robot stability. Scalar value r, is controlled by tuning the knee joint angle. Therefore, the distance between COM and ankle can be changed. Based on (23), F is a function of r and δ. As a consequence, r is proposed to be updated in

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

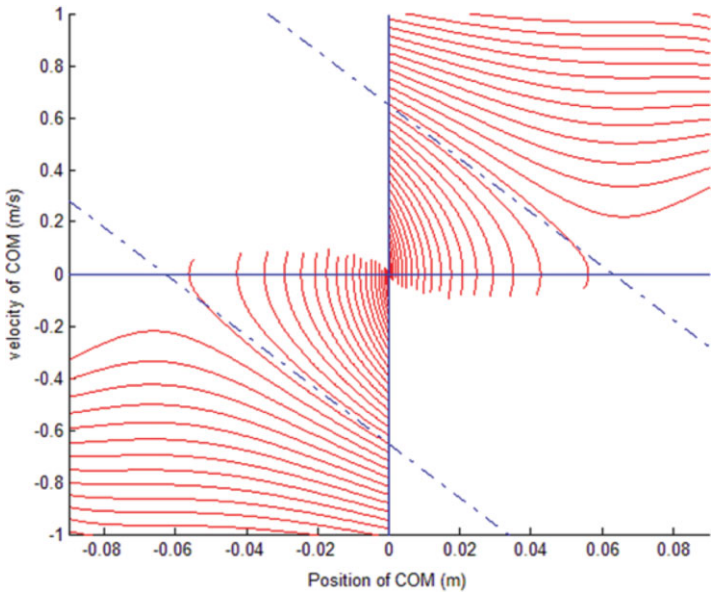


Fig. 6. (Colour online) Stable region of the knee strategy according to robo-builder specifications. The curves show how the robot can come back to stable region or fall.

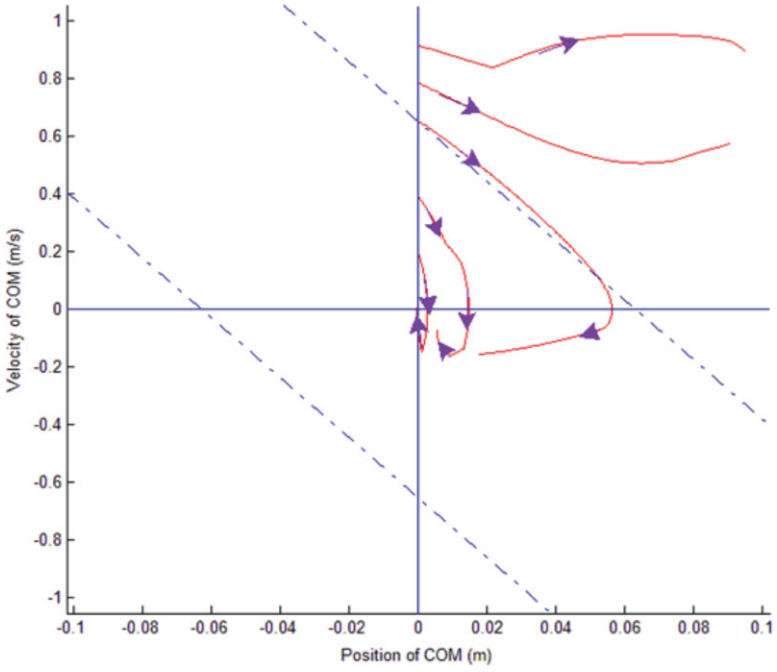


Fig. 7. (Colour online) Knee strategy trajectories which are controlled by minimizing the artificial potential error function. Gradient descent method is used to minimize the error function.

each step of gradient as followings:

$$r_{k+1} = r_k - \gamma \nabla F(r), \tag{25}$$

where $\nabla F\,(r) = \frac{\partial F}{\partial r}$ and $\gamma > 0$ is a small number. In case γ is chosen small enough, then

$$F(r_{k+1}) \leq F(r_k).$$

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

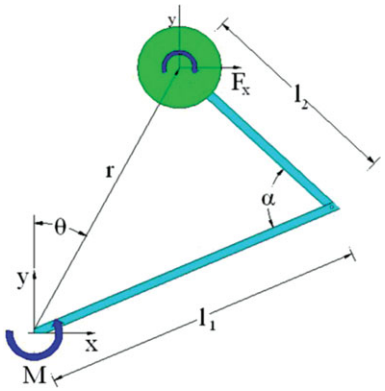


Fig. 8. (Colour online) Free body diagram of the robot in knee-hip strategy.

In order to find the relation between α, which is the knee angle, and r, the following equation and its derivative are used:

$$r^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos\alpha \Rightarrow \Delta r = \frac{l_1 l_2}{r} \sin\alpha \Delta\alpha, \tag{26}$$

where $\Delta r = r_{k+1} - r_k$. Therefore, by controlling the knee angle, r is updated in the way that F is decreased the most. δ, the other control parameter, is controlled by tuning the ankle joint torque. The following equation shows the relation between the ankle torque and δ through R_y:

$$\delta = \frac{\tau_{\text{ankle}}}{R_y}, \tag{27}$$

where R_y is illustrated in (20) and $\tau_{\text{ankle}} = \vec{\tau}_3$ which is denoted in (21). δ is updated in each step according to the gradient of F at δ as follows:

$$\delta_{k+1} = \delta_k - \gamma \nabla F(\delta). \tag{28}$$

The variation of the ankle torque is calculated based on the variation of the δ in each step. On the whole, knee strategy uses ankle torque and knee angle to control the robot stability and recover the balance of the robot against a push.

3.2. Knee-hip strategy

In previous subsection, knee strategy was explained in details. In this subsection knee-hip strategy is proposed based on knee strategy. Unlike the knee model, in knee-hip model, COM is considered as a flywheel. Indeed, knee-hip strategy is an extension of both knee and hip strategy.[1] Hip strategy was proposed by Stephens *et al.*[1] In hip strategy COM is considered as a flywheel. Stephens extended the model used by Pratt *et al.*[15] to propose the hip strategy.

Figure 8 illustrates the body diagram of the robot in knee-hip strategy. Knee-hip strategy has both advantages of the knee and hip strategy. Dynamics of the robot in knee-hip strategy is similar to knee strategy. Consider the maximum torque of the hip joint denoted by $\vec{\tau}_{\text{MF}}$, consequently, the dynamics of the robot can be rewritten as follows:

$$\vec{\tau}_1 + \vec{\tau}_2 + \vec{\tau}_3 \pm \vec{\tau}_{\text{MF}} = 0 \tag{29}$$

$$m(2r\dot{r}\dot{\theta} + r^2\ddot{\theta}) - mgr\sin\theta = mg\delta + m(\ddot{r} - r\dot{\theta}^2)\cos\theta\delta - m(2\dot{r}\dot{\theta} + r\ddot{\theta})\sin\theta \pm \vec{\tau}_{MF}. \tag{30}$$

More details on how (30) can be solved is available in appendix. By assuming the COM as a flywheel, the stable region of the knee-hip strategy can be extended in comparison with the knee strategy.

Figure 9 illustrates the stable region of the knee-hip strategy according to robo-builder specifications. Black parallel lines (..) show the stable region of the knee-hip strategy and blue

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

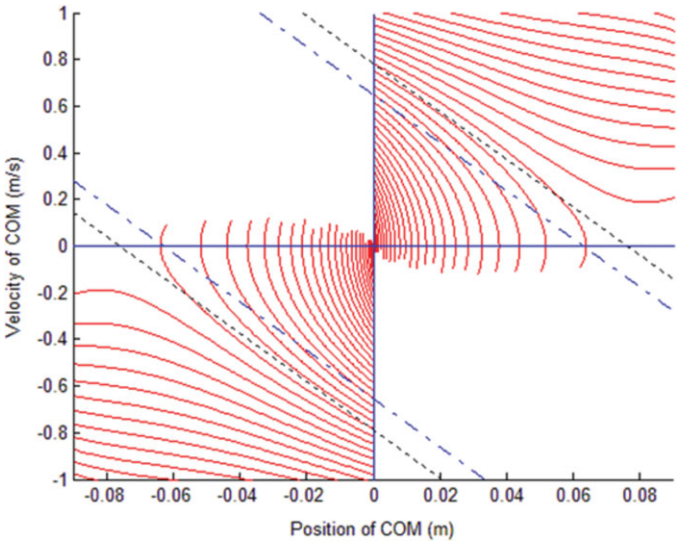


Fig. 9. (Color online) Stable region of the knee-hip. Black parallel lines (..): stable region of the knee-hip strategy. Blue parallel lines (-.): stable region of the knee strategy.

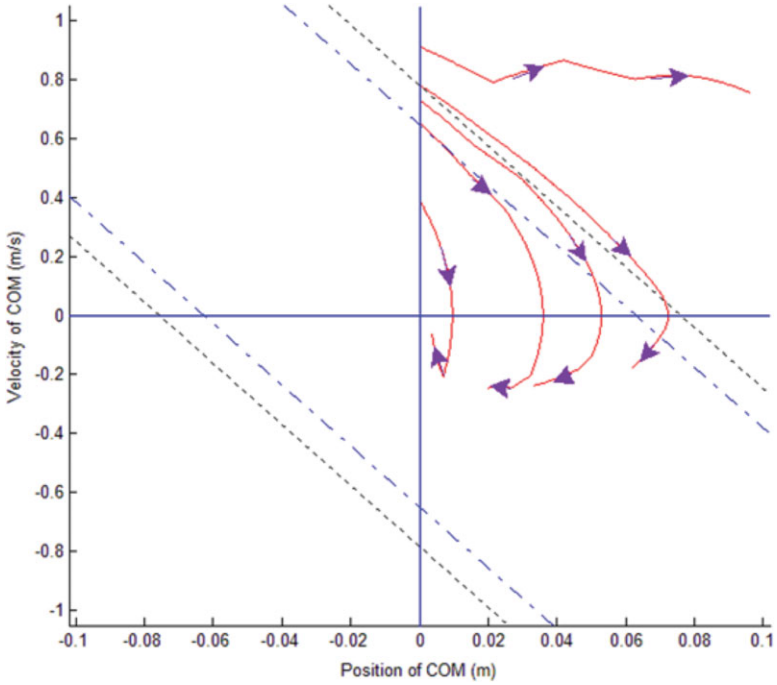


Fig. 10. (Colour online) Knee-hip strategy trajectories, which are controlled by minimizing the artificial potential error function. Gradient descent method is used to minimize the error function.

parallel lines (-.) illustrate the stable region of the knee strategy which was explained in details in the previous subsection. As it is obvious, knee-hip strategy, improves the stable region compared with knee strategy.

The flywheel torque should be applied in the opposite direction of the deviation angle to help the robot to recover its balance. Figure 10 illustrates how minimizing the artificial potential error function can control and move back the robot to the stable state. Those trajectories whose initial conditions are in the stable region of the knee-hip strategy converge to the stable state. Gradient descent method is used to minimize the artificial potential error function by controlling the ankle, knee, and hip joints.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

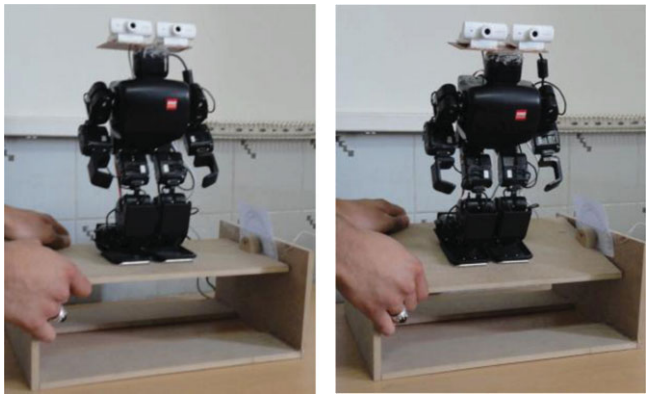


Fig. 11. (Colour online) Human robot is equipped with stereo cameras. The deviation is set manually and the platform deviation angle is read through an installed sensor.

The potential error function which is used in Knee-hip strategy is similar to knee strategy. Hip torque is the additional term which is added in the knee-hip strategy and it will be considered as a direct control variable.

4. Experimental Results

Experiments of the proposed method are conducted on a kid-size humanoid robot. This robot is assembled using a Robo-Builder kit and two cameras. The experiments are divided into two subsections. First subsection is the comparison between the vision-based estimations of the deviation angle, acceleration sensor based estimation of the deviation angle, the combined deviation angle, and the platform measured deviation angle of the robot. Second subsection compares the stable region of different strategies and illustrates a simple experiment of the knee strategy on the developed Robo-Builder platform.

To validate the output of the proposed method, a comparison was performed between the vision-based proposed method, output of the acceleration sensor, the combined, and the platform measured deviation angle of the robot. This deviation angle is obtained through a designed device which can be seen in Fig. 11 under the robot feet.

4.1. Vision-based estimation of the deviation angle of the robot

The platform deviation angle is read through an angle reader, a transducer, which is installed on the device. This sensor can read the deviation angle of the floor plate which is under the feet of the robot.

A fusion algorithm is proposed to combine the vision and acceleration sensor outputs. The proposed fusion algorithm is inspired from the well-known robust averaging algorithm.[18] Consider $S1$ and $S2$ as two signals. The combined signal, S_f, of $S1$ and $S2$ can be calculated as follows:

$$S_f(t) = \frac{\alpha(t)S1(t) + \beta(t)S2(t)}{\alpha(t) + \beta(t)}, \tag{31}$$

where α and β should be chosen proportional to inverse of the variance of $S1$ and $S2$ respectively. In our experiments, α and β are considered as follows:

$$\alpha(t) = \frac{1}{w} \sum_{\tau=-\frac{w}{2}}^{\frac{w}{2}} \left[S1\left(t-\tau\right) - \mu(t)\right]^2, \mu\left(t\right) = \frac{1}{w} \sum_{\tau=-\frac{w}{2}}^{\frac{w}{2}} S1\left(t-\tau\right). \tag{32}$$

Also, $\beta\left(t\right)$ is calculated similar to $\alpha(t)$ and according to $S2$ in each time step. w is the window size which is used to calculate the variances of the signals. Changing the window size affects the results. Fusion process of vision and acceleration sensor is done in different window sizes. The combined signal is compared with the other signals and the results are reported later.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

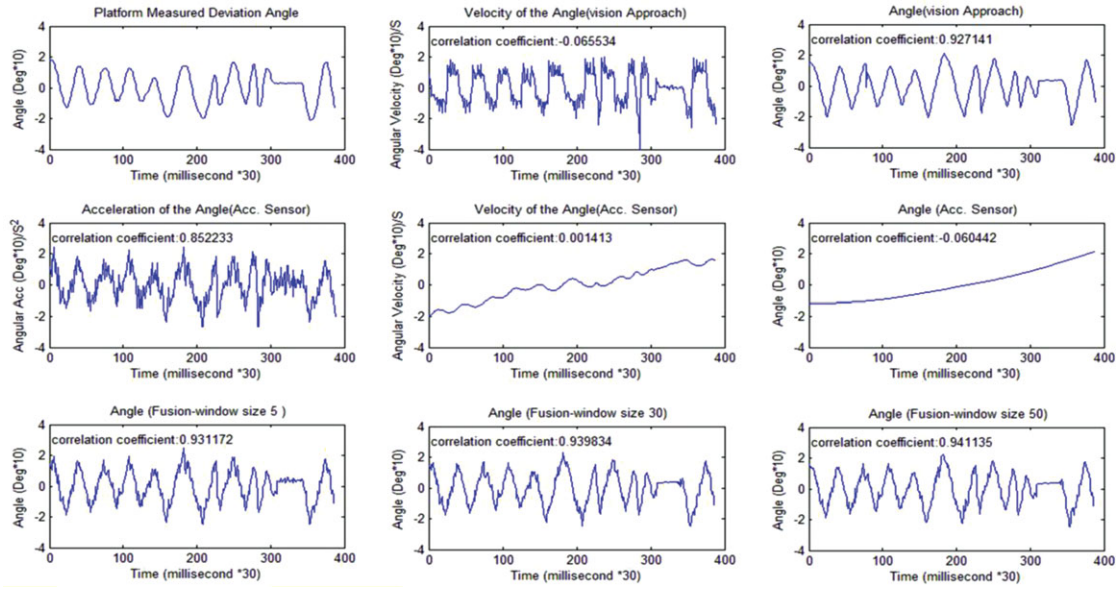


Fig. 12. (Colour online) Comparison between the platform measured deviation angle, velocity of the angle using vision technique, angle using the vision technique, acceleration obtained using acceleration sensor, velocity obtained using acceleration sensor, angle obtained using acceleration sensor, and combined deviation angle with different window size.

The source code of the vision-based method is written in C++ language and several open source libraries such as *opencv* and *openmp* are used in the implementation. The program was tested on a laptop with the following specifications: CPU: 2.0 GHz-Core2Dua, RAM: 3 GB DDR III. The vision program runs with 15–30 tracking pixels in the robotic laboratory environment. The runtime for each dual-frame including 2-D tracking and distance estimating is about 30 ms which can guarantee real time execution of the whole method.

Figure 12 illustrates the comparison of the platform measured deviation angle, estimation of the velocity of the angle using vision technique, estimation of the angle using the vision technique, acceleration obtained through acceleration sensor, velocity obtained through acceleration sensor, the angle obtained through acceleration sensor, and the combined signal of the vision angle and acceleration measured by acceleration sensor in different window sizes. Correlation coefficient is used as the criterion to evaluate the different methods. To evaluate each signal, the correlation coefficient of the platform measured deviation angle and the signal is calculated. Consider s1 and s2 are two signals, and cov(s1,s2) is the covariance of the signal s1 and s2. The similarity of s1 and s2 can be obtained through correlation coefficient,$C(s1, s2)$, as follow:

$$C(s1, s2) = \frac{\text{cov}(s1, s2)}{\sqrt{\text{cov}(s1, s1) * \text{cov}(s2, s2)}} \tag{33}$$

It is obvious in Fig. 12 that the deviation angle which is estimated by vision technique is similar to the platfrom deviation angle rather than acceleration sensor measurement. The acceleration module, which is used in experiments, is RBX-ACL3A01. This module is completely compatible with Robo-Builder kit. The deviation angle which is obtained using the fusion method performs suitably and it has the highest correlation coefficient among all the other signals. Although vision-based approach is smoother, combined signal has the highest correlation coefficient. In fact, fusion method considers lower weight for the signal which has a higher variance. For any reason, if a signal is measured noisy in a short time, fusion method considers a low weight for it. By changing the weight of signals in time, fusion can roughly handle the measurement errors. Fusion method performs the best at large window sizes.

The result of comparison between the platform measured deviation angle and the estimated deviation angle using vision technique is illustrated in Fig. 13.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

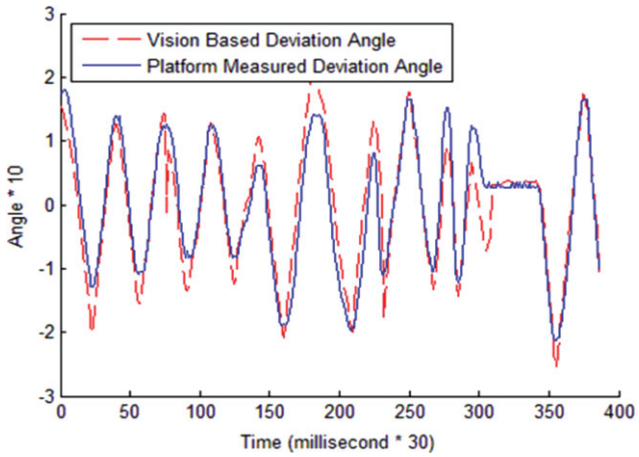


Fig. 13. (Color online) Red line (–): shows the vision-based estimation of the deviation angle, average of tracking 10 points; and the blue line (_ _): shows the platform measured deviation angle.

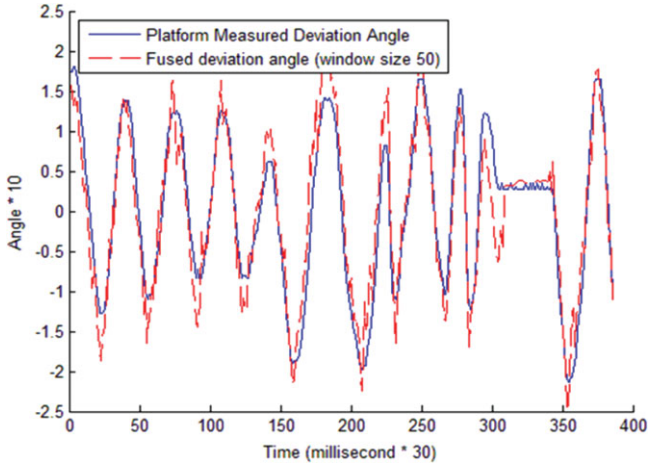


Fig. 14. (Color online) Red line (–): shows the combined estimation of the deviation angle, average of tracking 10 points; and the blue line (_ _): shows the platform measured deviation angle.

Indeed, this figure focuses on comparison between the platform deviation angle and the vision-based estimation of the deviation angle. Red line (dashed line) shows the deviation angle variations of the robot, which is estimated by the proposed vision-based method and the blue (solid) line shows the deviation angle which is measured using the installed sensor on the device. As it can be seen in Fig. 13, the proposed vision-based technique can estimate the deviation angle appropriately and it can be applicable in the real environment. In Fig. 13, the correlation coefficient of the platform measured deviation angle and the vision-based deviation angle is 0.92 which illustrates a high similarity between signals.

Figure 14 illustrates the result of comparison between the combined deviation angle and the platform measured deviation angle. As it is obvious the combined deviation angle converges to the platform measured deviation angle. The correlation coefficient of the platform measured deviation angle and the combined deviation angle is 0.94 which illustrates a high similarity between signals. By the point of view of the correlation coefficient, combined method performs better than vision method.

Table I illustrates the comparison between the vision-based deviation angle, the acceleration sensor output, and the combined deviation angle for eight experiments which are conducted in different duration time, number of tracking pixels, etc. The results have shown that vision-based estimation has better outcome compared with the acceleration sensor output. Also it is concluded that combining the acceleration sensor output and vision-based estimation could be more accurate.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

Table I. Comparison between vision-based estimation of the deviation angle and the acceleration sensor output.

	Vision-based estimation correlation	Combined Estimation correlation window size = 50	Acceleration sensor output correlation
Mean	0.93	0.94	0.82
Best	0.97	0.97	0.78
Worst	0.88	0.89	0.85

4.2. Stable region of push recovery strategies

The performance of the knee and knee-hip strategy is evaluated in this subsection by comparing them with other strategies.

The equation of the dynamics of the robot in knee and knee-hip models are nonlinear ordinary differential equation and cannot be solved or controlled precisely. Therefore, in this research this equation is solved numerically for various initial conditions (see the Appendix for more details). The properties of the robot that are utilized to solve the equations in all of the experiments in this paper are as follows:

$$l_1 = l_2 = 6\,\text{cm},\ \theta_i = 100^\circ, \dot{\theta} = 1 - 400^\circ/\text{s},\ \delta^{\pm}_{\max} = 5.1\ \text{cm},\ \vec{\tau}_{\text{MF}}(\max\ \text{hip torque}) = 0.11\ \text{m/kg}.$$

In which l_1 and l_2 are shank and thigh length of the robot respectively, θ_i is the initial knee angle, $\dot{\theta}$ is the angular velocity of knee joint, and $\vec{\tau}_{\text{MF}}$ is the maximum torque which robot can apply on hip joint.

The stable regions of the knee and knee-hip strategies are compared with the stable regions of the ankle and hip strategies presented by Stephens.[1]

Stephens proposed ankle and hip strategies and determined their stable regions. In ankle strategy, the robot is considered as a LIPM and the dynamics of his model was written as follows:

$$\ddot{x}_{\text{com}} = \frac{\text{g}}{\text{L}}(x_{\text{com}} - \delta^{\pm}), \tag{34}$$

where $\ddot{x}_{\text{com}}$ is the acceleration of COM, x_{com} is the position of COM, δ^- and δ^+ are the back and front edges of support region of the foot, L is the height of COM, and g is the acceleration due to the gravity.

Comparison between the correlation coefficient of the vision-based deviation angle, combined deviation angle with window size 50, and the output of the acceleration sensor. The results are for eight experiments which are carried out in different duration time, number of tracking pixels, etc.

Stephens also presented another strategy, which is called hip strategy. In hip strategy, COM is considered as a flywheel and the robot is considered as LIPM.

The dynamics of the robot in hip strategy is obtained by Stephens as follow:

$$\ddot{x}_{\text{com}} = \frac{\text{g}}{\text{L}}\left(x_{\text{com}} - \delta^{\pm} \pm \frac{\tau_{\max}}{mg}\right), \tag{35}$$

where $\tau_{\max}$ is the maximum torque which robot can apply on the hip joint and m is the mass of the robot.

Figure 15 illustrates the comparison of stable regions of the ankle, knee, hip, and knee-hip strategies. In Fig. 15 R1 shows the stable region of the ankle strategy. Ankle strategy has the smallest stable region among all strategies. R2 illustrates the improvement of the hip strategy compared with the ankle strategy. R3 demonstrates the performance of the knee strategy. It can extend the stable region of the hip strategy a little. R4 presents the stable region extension of the knee-hip strategy. Out of the stable region of the knee-hip strategy, robot cannot recover its stability through the mentioned strategies. Consequently, stepping can help the robot to avoid falling.

Stable regions of four mentioned strategies, which are explained and illustrated in Fig. 15, are obtained according to the robo-builder properties and it may differ a bit in other humanoids.

To evaluate the research in real environment, the stereovision approach and the proposed strategies were implemented and tested on the developed Robo-Builder platform. More document and details of the implementation such as video clips and photos are available on “www.mahani.info/Omid”.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

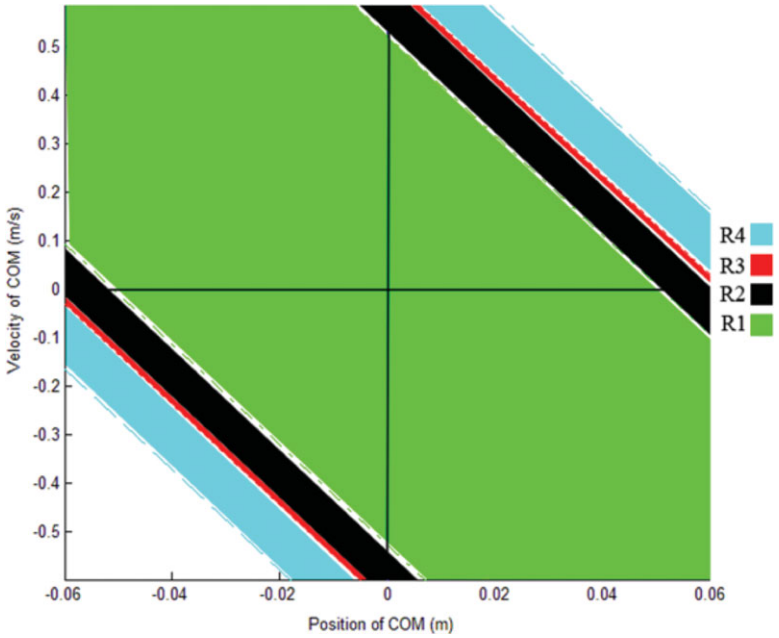


Fig. 15. (Colour online) Stable regions of different strategies. R1: ankle strategy, R2: hip strategy, R3: knee strategy, R4: knee-hip strategy.

5. Conclusion

Analyzing the COM motions is one of the best techniques for humanoid push recovery. The paper proposed a new approach for estimating the deviation angle of the robot and consequently, the COM motions based on the visual techniques. The proposed vision-based method used optical flow and stereovision techniques to estimate the deviation angle. Pyramid implementation of the optical flow is used to monitor two-dimensionally the tracking points, which are found according to an improved version of Harris corner detection, Shi-tomasi algorithm. Stereovision technique finds the distance of the tracking points. Since the stereovision works only on tracking points, the implementation of the proposed stereovision algorithm runs at frame rate of ordinary cameras.

By ignoring the LIPM assumption, knee and knee-hip strategies improve the stable region of the push recovery. In the proposed strategies, knee is considered as an effective joint and it influences the recovery process. Experiments illustrate that knee and knee-hip strategies can improve the push recovery regions, which are obtained by Stephens *et al.* By using knee and knee-hip strategies, robot can recover the stability against larger disturbances. Therefore, robot is not forced to take a step to recover the stability against larger pushes and it can use knee and knee-hip strategies instead. Experiments have been conducted on a humanoid robot and illustrate that the proposed method can be applied in real environments. Results show that knee strategy can recover the balance of the robot much better than ankle strategy, and knee-hip strategy can perform better than hip strategy.

Acknowledgements

The authors wish to thank R. Sameni, H. Sharifi, F. Tajeri-pour, B. Ranjbar, and L. Zarban for their helps during the research.

Appendix

Proof for the (3) and (4) according to Fig. 1 are as follows:

$$L_{1=}2d\tan\frac{\varphi}{2},\ \Delta = h\sin\theta,\ OB = d+\Delta = d+h\sin\theta$$

$$\omega = \theta - \frac{\varphi}{2},\ \gamma = \theta-\beta,\ \tan\beta = \frac{BP}{AB},\ \tan\omega = \frac{BC}{AB},\ \tan\theta = \frac{BD}{AB},$$

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

$$\cos\omega = \frac{AB}{AC},\ \cos\frac{\varphi}{2} = \frac{AE}{AC},\ \sin\frac{\varphi}{2} = \frac{EC}{AC}$$

$$x_P = \frac{X_P - h}{L_1} l \Rightarrow X_P = x_P \frac{L_1}{l} + h.$$

$$L_2 = 2EC = 2AC\sin\frac{\varphi}{2} \Rightarrow$$

$$L_2 = 2AB\frac{\sin\frac{\varphi}{2}}{\cos\omega} = 2(d + h\sin\theta)\frac{\sin\frac{\varphi}{2}}{\cos\left(\theta - \frac{\varphi}{2}\right)}.$$

$$\tan\gamma = \frac{EP'}{AE} \Rightarrow EP' = AE\tan\gamma = AC\cos\frac{\varphi}{2}\tan(\theta - \beta) \Rightarrow EP' = AB\frac{\cos\frac{\varphi}{2}}{\cos\omega}$$

$$\tan(\theta - \beta) = (d + h\sin\theta)\frac{\cos\frac{\varphi}{2}\tan(\theta - \beta)}{\cos\left(\theta - \frac{\varphi}{2}\right)}$$

$$x_{P'} = \frac{EP'}{L_2} l = \frac{\tan(\theta - \beta) l}{2\tan\frac{\varphi}{2}},\ \tan(\theta - \beta) = \frac{\tan\theta - \tan\beta}{1 + \tan\theta\tan\beta}$$

$$\tan\beta = \frac{X_P - X_B}{d + h\sin\theta},\ X_B = h\cos\theta,\ \tan\beta = \frac{2d\tan\frac{\varphi}{2}x_P + h(1 - \cos\theta)l}{(d + h\sin\theta)l}.$$

For solving the (23) or (30) the following method can be used:

$$(2r\dot{r}\dot{\theta} + r^2\ddot{\theta}) - gr\sin\theta + g\delta + (\ddot{r} - r\dot{\theta}^2)\cos\theta\delta - (2\dot{r}\dot{\theta} + r\ddot{\theta})\sin\theta\delta = 0$$

$$\Rightarrow (r^2 - r\sin\theta\delta)\ddot{\theta} = -2r\dot{r}\dot{\theta} + gr\sin\theta - g\delta - (\ddot{r} - r\dot{\theta}^2)\cos\theta\delta + 2\dot{r}\dot{\theta}\sin\theta\delta$$

$$\Rightarrow \ddot{\theta} = \frac{-2r\dot{r}\dot{\theta} + gr\sin\theta - g\delta - (\ddot{r} - r\dot{\theta}^2)\cos\theta\delta + 2\dot{r}\dot{\theta}\sin\theta\delta}{(r^2 - r\sin\theta\delta)}$$

$$\{Z\} = \begin{Bmatrix}\theta\\ \dot{\theta}\end{Bmatrix} \Rightarrow \{\dot{Z}\} = \begin{Bmatrix}\dot{\theta}\\ \ddot{\theta}\end{Bmatrix}$$

$$= \begin{bmatrix}\dot{\theta}\\ \frac{-2r\dot{r}\dot{\theta} + gr\sin\theta - g\delta - \left(\ddot{r} - r\dot{\theta}^2\right)\cos\theta\delta + 2\dot{r}\dot{\theta}\sin\theta\delta}{\left(r^2 - r\sin\theta\delta\right)}\end{bmatrix}.$$

References

1. B. Stephens, Control of Full Body Humanoid Push Recovery Using Simple Models, *Ph.D. Thesis* (Proposal, Carnegie Mellon University, Robotics Institute, 2009).
2. D. Tlalolini, C. Chevallereau and Y. Aoustin, "Human-like walking: Optimal motion of a bipedal robot with toe-rotation motion," *IEEE/ASME Transactions on Mechatronics*, **16**(2), 310–320 (2011).
3. M. A. Nikouei Mahani, S. Jafari and R. Rahmatkhah, "Novel Humanoid Push Recovery Using Knee Joint", *Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Bled, Slovenia (Oct. 26–28, 2011) pp. 446–451.
4. A. H. Adiwahono, Chee-Meng Chew, Weiwei Huang and Van Huan Dau, "Humanoid Robot Push Recovery Through Walking Phase Modification," *IEEE Conference on Robotics Automation and Mechatronics (RAM)*, Singapore (28–30 June, 2010) pp. 569–574.
5. B. J. Stephens, and C. G. Atkeson, "Push Recovery by Stepping for Humanoid Robots with Force Controlled Joints", *Proceedings of the 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Nashville, TN, USA (Dec. 6–8, 2010), pp. 52–59.
6. S. Kagamit, K. Nishiwakit, J. J. Kuffner Jr.t, Y. Kuniyoshi, M. Inabat and H. Inouet, "Online 3D Vision, Motion Planning and Bipedal Locomotion Control Coupling System of Humanoid Robot: H7," *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland (Sept. 30–Oct. 4, 2002) pp. 2557–2562.
7. S. Kanzaki, K. Okada and M. Inaba, "Bracing Behavior in Humanoid Through Preview Control of Impact Disturbances," *Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots*, Tsukuba, Japan (Dec. 5–7, 2005), pp. 301–306.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press

8. A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real time single camera SLAM", *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(6), 1052–1067 (2007).
9. S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear Inverted Pendulum Mode: A Simple Modeling for a Biped Walking Pattern Generation," *Proceedings of the IEEE/RSJ, Int. Conf. on Intelligent Robots and Systems*, Maui, Hawaii (2001) pp. 239–246.
10. H. Young-Dae, L. Bum-Joo, and K. Jong-Hwan, "Command State-Based Modifiable Walking Pattern Generation on an Inclined Plane in Pitch and Roll Directions for Humanoid Robots", *IEEE/ASME Trans. Mech.* **16**(4), 783–789 (2011).
11. K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Online Generation of Humanoid Walking Motion Based on a Fast Generation Method of Motion Pattern that Follows Desired ZMP," *Proceedings IEEE/RSJ Int. Conf. on Intelligent Robots and System EPFL*, Lausanne, Switzerland (Sept. 30–Oct. 4, 2002) pp. 2684–2689.
12. K. Yamamoto and Y. Nakamura, "Switching Feedback Controllers Based on the Maximal CPI Sets for Stabilization of Humanoid Robots", *Proceedings of the 9th IEEE-RAS international Conference on Humanoid Robots*, Paris, France (Dec. 7–10, 2009) pp. 549–554.
13. A. Hofmann, Robust Execution of Bipedal Walking Tasks from Biomechanical Principles, *Ph.D. Dissertation* (Massachusetts Institute of Technology, January 2006).
14. J. Pratt and R. Tedrake, "Velocity Based Stability Margins for Fast Bipedal Walking," *Presented in First Ruperto Carola Symposium in the International Science Forum of the University of Heidelberg entitled "Fast Motions in Biomechanics and Robots"*, Heidelberg, Germany (Sept. 7–9, 2005).
15. J. Pratt, J. Carff, S. Drakunov and A. Goswami, "Capture Point: A Step Toward Humanoid Push Recovery," *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots*, Genoa, Italy (Dec. 4–6, 2006) pp. 200–207.
16. X. Zeyang, X. Jing and C. Ken, "Global navigation for humanoid robots using sampling-based footstep planners", *IEEE/ASME Transactions on Mechatronics* **16**(4), 716–723 (2011).
17. J. Rebula, F. Canas, J. Pratt and A. Goswami, "Learning Capture Points for Humanoid Push Recovery", *Proceedings of the 7th IEEE-RAS International Conference on Humanoid Robots*, Pittsburgh, Pennsylvania (Nov. 29–Dec. 1, 2007) pp. 65–72.
18. J. M. Leski, "Robust Weighted Averaging of Biomedical Signals", *IEEE Transaction on Biomedical Engineering* **49**(8), 796–804 (2002).
19. A. Pretto, E. Menegatti, M. Bennewitz, W. Burgard and E. Pagello. "A Visual Odometry Framework Robust to Motion Blur", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe (May 12–17, 2009) pp. 2250–2257.
20. Jianbo Shi and Carlo Tomasi, "Good Features to Track", *Proceedings of the IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, Seattle, WA, USA (June 21–23, 1994) pp. 593–600.
21. C. Harris and M. Stephens, "A Combined Corner and Edge Detector", *Proceedings of the 4th Alvey Vision Conference*, Manchester, UK (Aug. 31–Sep. 2, 1998) pp. 147–151.
22. S. Denman, C. Fookes and S. Sridharan, "Improved Simultaneous Computation of Motion Detection and Optical Flow for Object Tracking," *Digital Image Computing: Techniques and Applications*, Melbourne, Australia (Dec. 1–3, 2009) pp. 175.
23. K. Kinoshita and K. Murakami, "Moving Object Tracking via One-Dimensional Optical Flow Using Queue," *Proceedings of the International Conference on Control, Automation, Robotics and Vision*, Hanoi, Vietnam (December 17–20, 2008) pp. 17–20.
24. Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker—Description of the algorithm", Intel Corporation – Microprocessor Research Lab.
25. S. Jafari and R. Jarvis, "Robotic eye-to-hand coordination: Implementing visual perception to object manipulation," *Int. J. Hybrid Intell. Syst. (IJHIS)* **2**(4) 269–293.
26. Su. Weijun, M. Rui and Yu. Chongchong, "A Study on Soccer Robot Path Planning with Fuzzy Artificial Potential Field", *Proceedings of the International Conference on Computing, Control and Industrial Engineering (CCIE)*, Wuhan, China (June 5–6, 2010) pp. 386–390.
27. K. B. Ngo, and R. Mahony, "Bounded Torque Control for Robot Manipulators Subject to Joint Velocity Constraints", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA*), Orlando, Florida (May 15–19, 2006) pp. 7–12.
28. L. Wachter, J. Murphy, and L. Ray, "Potential Function Control for Multiple High-Speed Nonholonomic Robots", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA*), Pasadena, California (May 19–23, 2008) pp. 1781–1782.

https://doi.org/10.1017/S0263574713000684 Published online by Cambridge University Press