

Biologically Inspired Kinematic Synergies Provide a New Paradigm for Balance Control of Humanoid Robots

Helmut Hauser ^{#1}, Gerhard Neumann ^{#2}, Auke J. Ijspeert ^{*3}, Wolfgang Maass ^{#4}

#Institute for Theoretical Computer Science

Graz University of Technology

Inffeldgasse 16b/I, A-8010 Graz, Austria

^{1,2,4}{hauser,neumann,maass}@igi.tugraz.at

**Ecole Polytechnique Fédérale de Lausanne (EPFL)*

School of Computer and Communication Sciences

Station 14, CH-1015 Lausanne, Switzerland

³auke.ijspeertg@epfl.ch

Abstract—Nature has developed methods for controlling the movements of organisms with many degrees of freedom which differ strongly from existing approaches for balance control in humanoid robots: Biological organisms employ kinematic synergies that simultaneously engage many joints, and which are apparently designed in such a way that their superposition is approximately linear. We show in this article that this control strategy can in principle also be applied to balance control of humanoid robots. In contrast to existing approaches, this control strategy reduces the need to carry out complex computations in real time (replacing the iterated solution of quadratic optimization problems by a simple linear controller), and it does not require knowledge of a dynamic model of the robot. Therefore it can handle unforeseen changes in the dynamics of the robot that may arise for example from wind or other external forces. We demonstrate the feasibility of this novel approach to humanoid balance control through simulations of the humanoid robot HOAP-2 for tasks that require balance control under disturbances by unknown external forces.

I. INTRODUCTION

Balance control for humanoid robots is known to be a difficult problem because of the large number of degrees of freedom (DoF) that are involved. It has been simplified through the discovery that it suffices for balance control to keep the center of pressure (CoP) (or related points on the ground [1]) within the 2-dimensional support polygon defined by the convex hull of those points where the feet are supported by the ground. Powerful computational methods have been developed for achieving this (see e.g. [2], [3]) by solving suitable optimization problems. While [2] manages to achieve in this way even online balance compensation (by solving a quadratic programming problem), these approaches require knowledge of the precise dynamic model of the robot, and therefore cannot be applied in situations where the dynamic model of the robot may change online, due to wind, picking up loads, contact with other robots, or other external forces.

Biological organisms face similar problems, in fact, the dynamic model of their movement apparatus changes in

addition during development, but also through injuries. But the available experimental data suggest that biological organisms employ a radically different strategy for controlling a movement apparatus with many DoF, in particular for balance control. Numerous studies from the Lab of Bizzi at MIT [4], [5], [6] have shown that the central nervous systems of a variety of organisms employ a modular architecture for motor control, whereby many different movements (arm movements, walking, jumping, swimming) can be constructed as largely linear (but non-negative) combinations of a rather small repertoire of movement primitives (also referred to as muscle synergies, or kinematic synergies; we use the latter term in this article). Figure 4 in [6] shows that kinematic synergies in frogs engage largely disjoint sets of muscles, which may facilitate an approximate linearization of movement control through non-negative linear combinations of these kinematic synergies. Also recent work on whole-body movements of humans [7], [8] shows that balance control and other human body movements during standing can be understood as combinations of a very small set of stereotypical kinematic synergies, that each affect several joints.

We explore in this article the question whether an analogous modularization and simplification strategy might provide a viable alternative to existing balance control methods in humanoid robots. The first problem that one faces here is the design of suitable kinematic synergies. In biological organisms they are assumed to result from a combination of genetic encoding and developmental learning. For humanoid robots one could apply PCA to recordings of the movements of multiple joints of humans (this is the way in which kinematic synergies were extracted in [7], [8]) or one can use an even simpler strategy that we describe in section (III), and apply to the humanoid robot HOAP-2. We show in section (IV) how the resulting two kinematic synergies for moving the CoP left-right and forward-backwards can be used to design a control loop with a linear controller for balance control of

the HOAP-2. We show in section (V) that a simple linear controller is able to compensate for random movements of a surfboard on which the robot stands, even in the presence of unforeseeable online changes in the dynamic model of the robot and its environment. Additionally we demonstrate that the proposed framework can be used to follow any desired trajectory for the CoP.

II. FORMAL DEFINITION OF KINEMATIC SYNERGIES

We regard a kinematic synergy as a fixed number m of degrees of freedom (joint angles) which cooperate under the regime of a one dimensional input parameter s . It can be represented as a nonlinear function Φ depending on one controlling parameter $s \in \mathbb{R}$.

Definition 1: A kinematic synergy (KS) is a function $\Phi := \Phi(s)$ which maps a parameter $s \in \mathbb{R}$ onto a m dimensional vector of joint angles $\mathbf{q}^{KS} = \Phi(s)$:

$$\Phi : \mathbb{R} \rightarrow \mathbb{R}^m . \quad (1)$$

The superscript KS denotes the subset of m joints of the robot, which are controlled by the KS. The total number of joints in the robot is denoted by n . Further we define the function φ

$$\varphi : \mathbb{R}^m \rightarrow \mathbb{R}^n \quad (2)$$

which embeds the m -dimensional subspace spanned by Φ into the n -dimensional space of all joints of the robot. This embedding copies the positions for all joints affected by Φ and leaves the remaining joints constant.

A KS is typically applied in order to control a low-dimensional, or even one-dimensional, variable y . In this article we use two KS's that each control one dimension of the center of mass projected to the ground (PCoM). Since the PCoM depends on the posture of the whole robot, the output y depends on all n joint positions $\mathbf{q} \in \mathbb{R}^n$ of the robot. Generally speaking we have a nonlinear function \mathbf{f} depending on n degrees of freedom

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R} \quad (3)$$

and we want the KS to control its corresponding output $y = \mathbf{f}(\mathbf{q})$.

Figure 1 shows the mapping of Φ from the one dimensional s -space to the m -dimensional joint space, the embedding with the function φ and the mapping of function \mathbf{f} from the n -dimensional space of all joints to the one dimensional output y .

Since such KS affects m degrees of freedom that depend just on a one dimensional parameter s , we can impose further constraints on the function Φ . A reasonable choice for such a constraint is a linear relationship between the controlling parameter s and its corresponding output y . This will reduce the nonlinearities inherent to any kinematic chain and hereby facilitate controlling and learning. Hence we are particularly interested in the following type of KS:

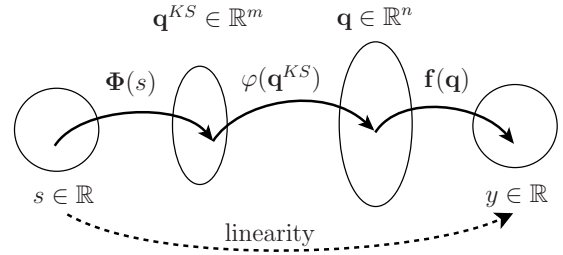


Fig. 1. Scheme for the composition of the functions φ and \mathbf{f} according to (2) and (3) with the kinematic synergies Φ .

Definition 2: A linearizing kinematic synergy is a kinematic synergy according to (1) such that there exists a linear relationship between its controlling parameter s and the corresponding (to be controlled) output y .

$$y = (\mathbf{f} \circ \varphi \circ \Phi)(s) = k \cdot s \quad (4)$$

for some $k \in \mathbb{R}$.

We restrict our attention in this article to such *linearizing* KS, to which we simply refer as KS.

In practice we deal with dynamics, and we want to control the robot to accomplish dynamical tasks. Let us take a closer look at why it is still an advantage to use kinematic synergies:

Let $\mathbf{q} \in \mathbb{R}^n$ be the positions (angles), $\dot{\mathbf{q}} \in \mathbb{R}^n$ be the velocities and $\ddot{\mathbf{q}} \in \mathbb{R}^n$ be the accelerations of all n joints of the robot. Then the state of the whole robot is defined as $\mathbf{p} = [\mathbf{q}^T, \dot{\mathbf{q}}^T, \ddot{\mathbf{q}}^T]^T$ and therefore $\mathbf{p} \in \mathbb{R}^{3n}$. Note that in general any dynamical task will include the whole state space. The KS just controls m of n joints and only maps on the joint positions $\mathbf{q}^{KS} \in \mathbb{R}^m$. By definitions (1) and (2) we restrict the joint space of m joints. Loosely speaking we just allow defined postures for the robot. This makes sense since most of the possible postures are not valid due to stability issues, crossing of limbs, etc. . If one assumes that the $i = n - m$ joints which are not dominated by Φ do not move at all, then we can neglect the embedding φ for the subsequent derivations. Any change $\frac{ds}{dt} := \dot{s}$ of the control parameter causes a corresponding change of joint positions $\dot{\mathbf{q}}^{KS}$

$$\dot{\mathbf{q}}^{KS} = \frac{\partial \Phi}{\partial t} = \frac{\partial \Phi}{\partial s} \cdot \frac{\partial s}{\partial t} = \frac{\partial \Phi}{\partial s} \cdot \dot{s} .$$

The same holds for the acceleration $\frac{d^2 s}{dt^2} := \ddot{s}$

$$\ddot{\mathbf{q}}^{KS} = \frac{\partial^2 \Phi}{\partial t^2} = \frac{\partial^2 \Phi}{\partial s^2} \cdot (\dot{s})^2 + \frac{\partial \Phi}{\partial s} \cdot \ddot{s} .$$

Therefore the velocities $\dot{\mathbf{q}}$ and the accelerations $\ddot{\mathbf{q}}$ of the joints are also restricted to a subspace. Obviously this strategy facilitates the control of the system.

III. APPLICATION TO THE HUMANOID ROBOT HOAP-2

We now test whether kinematic synergies provide an alternative to existing methods (see e.g. [2], [3]) for balance control of the humanoid robot HOAP-2 [9]. This robot has $n = 25$ degrees of freedom (rotational joints). Its structure

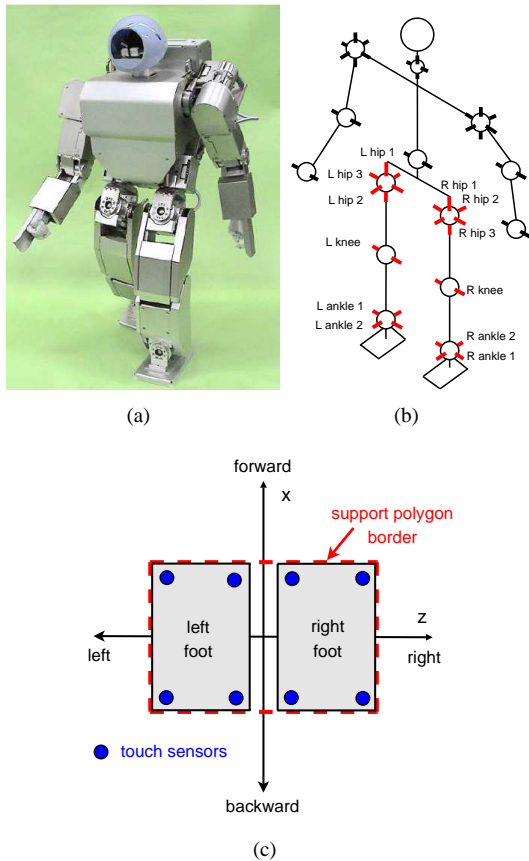


Fig. 2. (a) The real HOAP-2 robot and (b) its schematic structure. Red marked and labeled joint rotation axes are used for Φ . (c) Support polygon on the support surface for the robot, with touch sensors that are used to estimate the pseudo center of pressure ($pCoP$). Black arrows indicate the x dimension (forward/backward: range 9.5 cm) and z dimension (left/right: range: 14.3 cm) for movements of the center of pressure.

is shown in Figure 2(b). The task is to balance dynamically the HOAP-2 against external perturbations. Therefore a natural choice for the output function \mathbf{f} could be any dynamic stability point like the center of pressure (CoP), the zero moment point (ZMP) or the foot rotation indicator (FRI) [1], [10]. We want to control the humanoid robot standing in double support phase. In practice these stability points are estimated by pressure sensors. HOAP-2 has four such sensors per foot, located at the corners (see Figure 2(c)). We call the dynamic stability point that one computes from the outputs of these sensors the *pseudo center of pressure* ($pCoP$). The goal is to keep the $pCoP$ within the support polygon shown in Figure 2(c). Since a KS presents a static mapping we are going to use the static model (i.e. kinematic model and mass distribution) and the PCoM to obtain the KS . Note at zero joint velocities $\dot{\mathbf{q}}$ and zero joint accelerations $\ddot{\mathbf{q}}$ the PCoM coincides with the $pCoP$.

In one of our subsequently discussed experiments the robot stands on a platform. If this platform tilts, the support polygon moves with the platform, but the projection of the CoM and the $pCoP$ is no longer orthogonal to the support polygon (but orthogonal to some ideal horizontal plane). Hence the PCoM

and the $pCoP$ are likely to move within (and possibly beyond) the support polygon when the platform tilts. We will assume that the feet of the robot do not slip, and hence we can use the same stability criterion as for the case of a horizontal platform. The robot does not fall over as long as its $pCoP$ stays within the support polygon (even if the support polygon is now tilted).

In definition (2) we defined the output to be one dimensional ($y \in \mathbb{R}$). Since dynamic stability points correspond to points on the contact surface, and are therefore two dimensional, it is natural to define two KS 's Φ_x and Φ_z for corresponding output dimensions x and z . The output functions \mathbf{f}_x and \mathbf{f}_z correspond to the two dimensions x and z of the PCoM. The kinematic synergy Φ_x controls the movement forward/backward and the kinematic synergy Φ_z left/right relative to the robot (see Figure 2(c)).

The number of degrees of freedom controlled by the kinematic synergies is $m = 12$: three hip joints, one knee joint and two ankle joints for both legs (see Figure 2(b)).

Since $n > m$, not all joints of the robot are controlled by the kinematic synergies. The additional $i = n - m$ joints could be used for other tasks (grasping, waving, tracking objects, etc.). Their movement will change the outputs y_x and y_z to some extent, but will be interpreted by the kinematic synergies as an external perturbation. We will demonstrate in one experiment (see section V) that our approach is capable to deal with this type of disturbances too.

A. Obtaining Motion Primitives with Inverse Kinematics

The desired kinematic synergies are constructed via inverse kinematics. They are constructed just once for the robot and are fixed during control action. With the desired linearization property (4) one still has an infinite number of possible kinematic synergies. Therefore their construction can satisfy extra constraints which depend on the set of tasks the KS should accomplish. For our balancing task we want to assure upright posture for the upper body and double support (both feet have contact with the ground).

Based on the given kinematic model and the mass distribution of the HOAP-2 the kinematic synergies Φ_x and Φ_z have been calculated as follows: An initial posture \mathbf{q}_{init} has been defined which results (for the case of a horizontal support surface) in a PCoM at the center of the support polygon \mathbb{S} . By definition we set the origin of the coordinate system to the center of the support polygon and therefore the resulting outputs in the initial posture are $\mathbf{f}_x(\mathbf{q}_{init}) = \mathbf{f}_z(\mathbf{q}_{init}) = 0$. To reflect the natural limit of static stability the KS parameter s is normalized such, that -1 and $+1$ correspond to the borders of the support polygon \mathbb{S} . Therefore the region of acting without falling will be (for the case of a horizontal support surface) the range $s \in [-1, +1]$ for both dimensions x and z , see Figure 2(c).

The construction of each KS consist of two optimization steps, which we describe only for Φ_x . Similar steps lead to the second kinematic synergy Φ_z . We divided arbitrarily the range of the s_x -parameter over the support polygon into 80 points. Therefore each step was $\frac{9.5 \text{ cm}}{80} \approx 0.12 \text{ cm}$ and correspond to

a step of $\Delta s_x = 0.025$. From the starting posture \mathbf{q}_{init} the next posture \mathbf{q}' is calculated to get a desired output y'_x , which is one step (0.12 cm) away from the origin of the coordinate system in the x -direction.

The first optimization step, which used all 12 joints of the legs, was used to move the PCoM of the robot in the x -direction and simultaneously tried to keep the upper part of the body in an upright position. The coordinate system of the left leg was always used as origin, all positions and rotations of the remaining body parts were calculated relatively to the left leg's coordinate frame. In order to calculate the new joint positions, a Jacobian Pseudo-Inverse approach was used [11]. The applied Jacobian matrix consisted of two $3 \times n$ submatrices, the Jacobian for the CoM-Position and the Jacobian for the rotation of the torso. Therefore, both constraints, moving the CoM in the desired direction and keeping the upper part of the body upright were fulfilled by the optimization. The coordinate frame of the left foot was kept constant, but the position of the right foot tended to change, so that it was likely to loose ground contact.

Therefore the second optimization step was used to move the right foot back into its original position. For this optimization only the 6 joints relevant for the right foot were used and again a Jacobian Pseudo-Inverse approach was applied.

These two steps were iterated until the output value y'_x was reached. The resulting new posture \mathbf{q}' was then set to be the new starting point. The next posture \mathbf{q}'' was found by the same procedure. This approach was applied iteratively until the output y_x reached the border of the support polygon. The same procedure was applied for the opposite direction (i.e. s_x from 0 to -1). The process leads to a look up table for the range $s_x \in [-1, +1]$. Joint positions in between the steps, if needed, are calculated by linear interpolation.

Figure 3 shows the kinematic synergies Φ_x and Φ_z which result from this procedure. One can clearly see that (analogously as their biological prototypes, see Figure 4 in [6]) these two KS's primarily affect disjoint set of joints, thereby supporting linearity of superpositions. Figure 4(a) shows the mapping from the parameter s_x that controls the KS Φ_x to the coordinates $y_x = PCoM_x$ and $y_z = PCoM_z$ of the PCoM. Note the linear relationship, and that the other output dimension $y_z = PCoM_z$ is unaffected by s_x . Figure 4(b) shows the same for the KS Φ_z .

Since we have two kinematic synergies which are mapping to the same 12 joints, we are not able to use joint positions directly for the encoding of each KS. Therefore only joint offsets $\Delta \mathbf{q}^{KS}$ from the initial position \mathbf{q}_{init} are stored in the look up table. When both kinematic synergies are applied, the resulting target joint positions are calculated by summing up the initial position and the two offsets at every single joint. This is possible since the joints mainly responsible for the movement in x -direction are orthogonal to the joints mainly responsible for the z -direction (see Figure 3). Figure 5 shows the results of an empirical evaluation of the approximate linearity of the KS's Φ_x and Φ_z . Except for cases where both of the variables s_x and s_z reach their extremal values (these

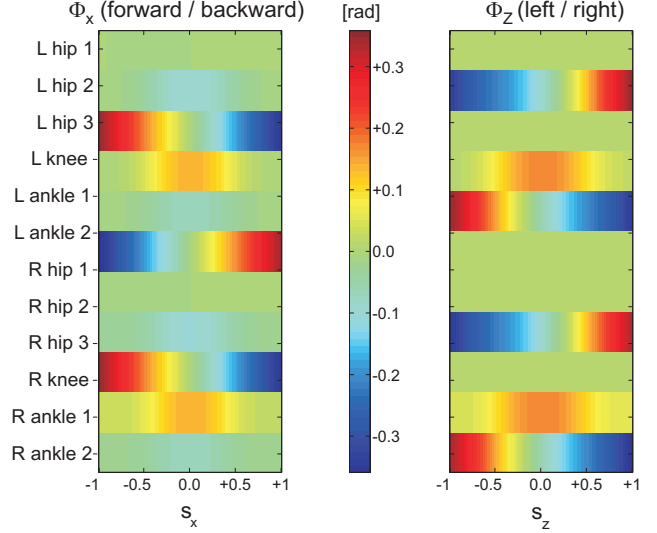


Fig. 3. Kinematic synergies Φ_x and Φ_z for the HOAP-2 that move the PCoM forward/backward and left/right over the range $[-1,+1]$ for the KS-parameters s_x and s_z . Note that these two KS's primarily affect disjoint sets of joints.

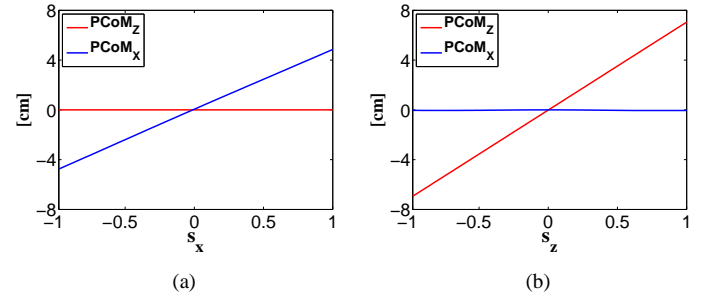


Fig. 4. (a) The KS Φ_x maps the control variable s_x linearly onto the x - and z -coordinates of the PCoM. It strongly affects the x -coordinates, but keeps the x -coordinates of the PCoM approximately constant. (b) The same holds for the second KS Φ_z .

are the cases when the PCoM is moved into a corner of the support polygon) the deviation from linearity is quite small.

IV. BALANCE CONTROL WITH KINEMATIC SYNERGIES

The kinematic synergies Φ_x and Φ_z were constructed to control the output function PCoM. However the robot can estimate from its pressure sensors in the feet only the $pCoP$, which is also affected by the dynamics of the robot. Similarly, values from simulated pressure sensors are used in our simulations of the HOAP-2 to estimate the PCoM.

Assumption: The robot moves sufficiently slowly so that one can assume that approximately

$$pCoP \approx PCoM . \quad (5)$$

We will show in section V that external forces that change on a time scale of seconds, such as wind, can be handled by movements of the HOAP-2 which satisfy this assumption. With this assumption we are able to control the humanoid

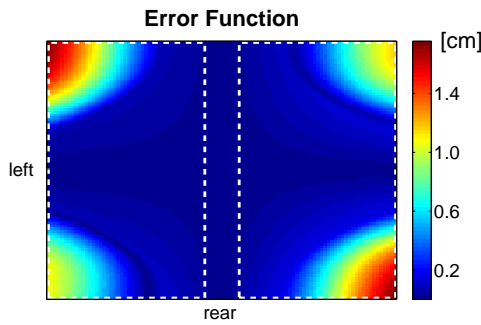


Fig. 5. Empirical evaluation of the approximate linearity of the superposition of the KS's Φ_x and Φ_z . For every point $\langle y_x, y_z \rangle$ in the support polygon we applied the KS's with values s_x, s_z that were supposed to produce a posture where the coordinates of the PCoM are $\langle y_x, y_z \rangle$ (based on the linearity assumption 2 for each KS, and the assumption of linearity of the superposition). The Euclidean distance between $\langle y_x, y_z \rangle$ and the resulting actual position of the PCoM is shown in color code. One clearly sees the error caused by nonlinearity is in the mm-range for most of the support polygon, and only becomes larger in the corners. The white dotted lines depict the contours of the feet.

robot with simple linear controllers in conjunction with the KS's. Note that even if this assumption does not hold, one can use KS's for balance control. But then, instead of a simple linear controller, a more sophisticated controller has to be applied.

We now show how the kinematic synergy Φ_x can be used to construct a linear control loop for balancing the robot in the x -dimension. The Φ_z is analogous. If the assumption holds, the function from the time derivative \dot{s}_x of the KS-parameter s_x to the value $pCoP_x$ of the estimated $pCoP$ in the x -dimension can be sufficiently represented by a linear transfer function of the form:

$$P(z) = \frac{K}{(z-1)} \text{ with } K \in \mathbb{R}^+, \quad (6)$$

with z being the time shift operator for discrete systems [12]. The denominator polynomial represents an integrator (one pole at $z = +1$) which integrates the velocity \dot{s}_x of the KS-parameter to obtain s_x .

By the assumption this approach has a natural limit, but as long as the dynamical effects are small enough they can be seen as uncertainties in the linear model. Already a simple linear controller can deal with these small uncertainties. To get a closed control loop with feedback we define a feedback error

$$e_x := \tilde{y}_x - y_x \quad (7)$$

with \tilde{y}_x being the desired output value and y_x being the corresponding value $pCoP_x$ that is computed from the pressure sensors. The goal is to prevent the robot from falling over. Therefore the $pCoP_x$ should stay close to the center of the support polygon \mathbb{S} . Since we have defined the center of \mathbb{S} at the origin, see Figure 2(c), the desired value is $\tilde{y}_x = 0$ (and $\tilde{y}_z = 0$ for the z -dimension).

Now we can formulate a standard PID-controller with the

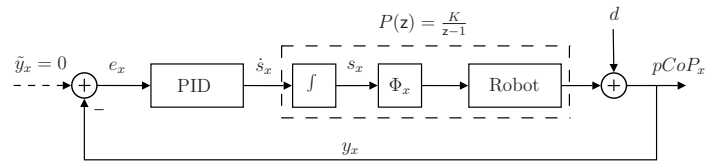


Fig. 6. Closed control loop for one kinematic synergy Φ_x . The reference point is set to $\tilde{y}_x = 0$ since we want to have the $pCoP_x$ at the center of the support polygon. External perturbation d is a change of $pCoP_x$ not introduced by the kinematic synergy control loop.

following equation for the controller output u_x :

$$u_x = K_P \cdot e_x + K_I \int e_x dt + K_D \frac{de_x}{dt} = \dot{s}_x \quad (8)$$

Figure 6 shows the closed control loop for one kinematic synergy (Φ_x). Note that the structure of the control loop is a standard feedback control loop which has the property to suppress perturbations d .

The formulation of the control loop is very general. The previously described control scheme is designed to control around an set point ($\tilde{y}_x = \tilde{y}_z = 0$). Of course one could use the same control loop to move the $pCoP$ on any desired trajectory (time varying $\tilde{y}_{x,z}(t)$) as we show later in an experiment. This is useful for example to initiate a walking cycle. The robot has to move the $pCoP$ under the future supporting foot in order to be able to raise the other leg without falling.

A. Controller Design

To find appropriate values for the PID-controller (8) we can use the rich set of tools which are offered by linear control theory. The values $K_P = 15$, $K_I = 0$ and $K_D = 0.0001$ we used, were empirically found to have a reasonable performance. There exist of course a number of possible improvements to get better controllers: adaptive control or robust control approaches, optimal control or different trial and error approaches to find good control parameters. Even higher order controllers or different control structures than in Figure 6 are possible. Here we just present a very simple and straightforward implementation with a PID controller to illustrate the new strategies for balance control which become feasible through the use of kinematic synergies.

B. Balance Control Task

We consider three tasks. First we investigate the capability of the approach under static conditions. Second we balance the humanoid robot HOAP-2 on a randomly moving platform (i.e. surfboard) with additional external forces (winds). Third we demonstrate the capability to follow a general CoP trajectory and to deal with movements of joints not under the regime of the KS's.

A detailed dynamical model of the HOAP-2 robot based on data from the vendor Fujitsu has been implemented in the robot simulation software Webots [13] and has been used for our simulations. The basic simulation time step has been set to 1 ms. The time step for the control has been set to 8 ms.

	x-direction		z-direction	
	forward	backward	left	right
without control	+10.6°	-8.6°	-14.3°	+14.3°
with control	+20.1°	-22.3°	-26.4°	+26.4°
improvement	89.6%	159.3%	84.6%	84.6%

TABLE I

THE FIRST TWO ROWS SHOW THE TILT ANGLE (OF THE PLATFORM ON WHICH THE HOAP-2 WAS STANDING) AT WHICH THE ROBOT LOOSES ITS BALANCE, BOTH WITHOUT A CONTROLLER WHICH CHANGES THE POSTURE OF THE ROBOT IN ORDER TO COMPENSATE FOR THE TILTED PLATFORM, AND WITH THE LINEAR CONTROLLER BASED ON KINEMATIC SYNERGIES THAT IS DESCRIBED IN SECTION IV. THE CONTROLLER ENABLES THE ROBOT TO TOLERATE ABOUT TWICE THE TILT ANGLE.

The surfboard can rotate about the x -axis with angle Θ_x and also about the z -axis with the angle Θ_z . Trajectories for Θ_x and Θ_z are generated independently for each dimension. These were found by following procedure: We just generated a random trajectory of jumps with random amplitude and random durations. In order to get a smooth trajectory we applied additionally a discrete smoothing FIR-filter with three poles at 0.997. Typical resulting trajectories are presented in Figures 7(a) and 7(b).

Since we have two kinematic synergies (Φ_x and Φ_z) we have two control loops. They react independently from each other on their corresponding output dimension x and z respectively. Both linear controllers calculate, depending on their errors e_x and e_z , the velocities \dot{s}_x and \dot{s}_z of their KS-parameters. These are integrated to s_x and s_z , which result via the look up table into joint angle offsets. They are summed up as described in section (III-A) to get the actual joint target positions. Local PD-controllers at the servos transform then the desired angles into torques.

V. RESULTS

In a first test of the capability of the linear controller based on KS's (that was described in the preceding section) we tilted the platform on which the HOAP-2 was standing, and determined at which tilt angle the robot falls over. This tilting was carried out very slowly in this first test, and separately for the two tilt axes that move the CoP in the x and z direction. Table I shows that our control strategy allows the robot to tolerate about twice the tilt angle without loosing its balance – compared with a robot which does not change its posture in an adaptive manner.

We next considered the case where the platform on which the robot stands is tilted dynamically in random directions according to the stochastic process described in the preceding section – as one might have for example on a surfboard – (see Figure 7(a), 7(b)), and the previously described controllers based on kinematic synergies for the x - and z -direction are simultaneously active. In addition random external forces (which might for example arise from wind or contact with other objects) were applied to the CoM (at the torso) of the robot at various points in time. These external forces change the dynamic model of the robot in an online manner, and therefore make control strategies that require knowledge of

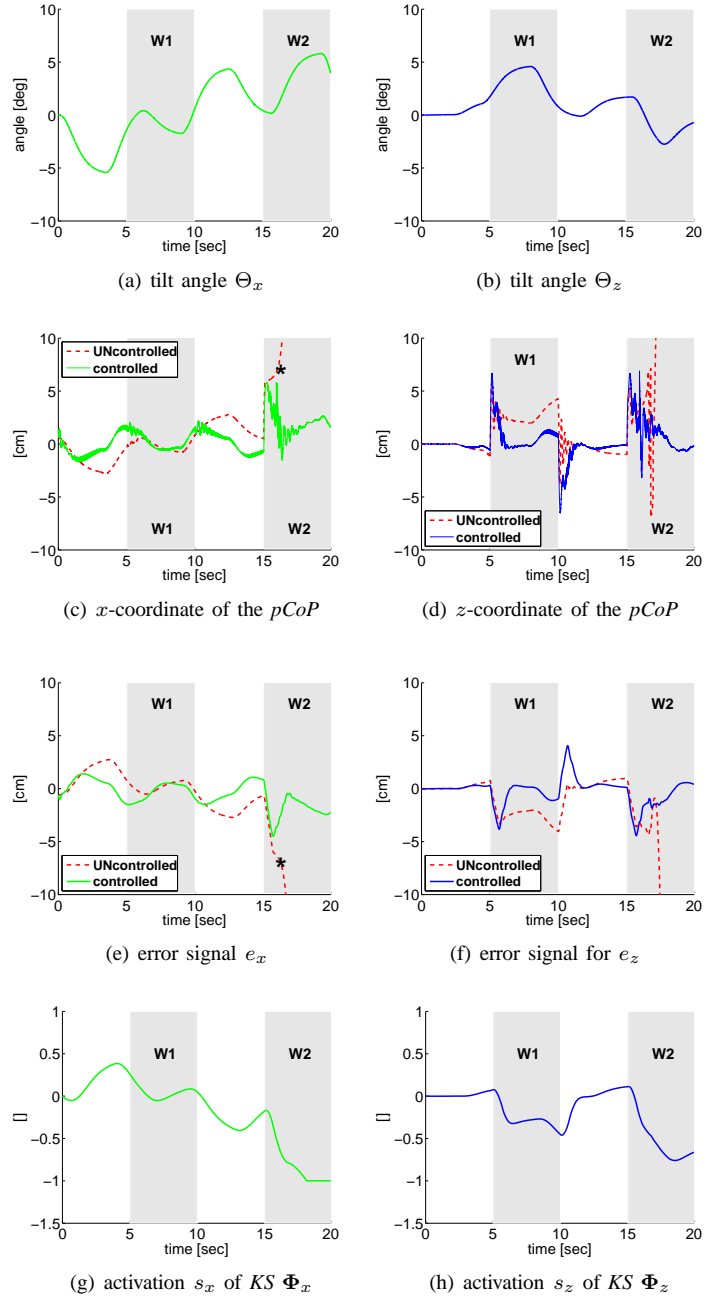


Fig. 7. Result from the test with a moving support platform ("surfboard") and unexpected external forces ("wind") $W1$ and $W2$. The balance of the HOAP-2 is controlled by two linear controllers, based on two kinematic synergies. Without balance control (red dashed line in (c) and (d)) shows trajectory of $pCoP$ without active balance control) the $pCoP$ would leave the support polygon at time point 16 s (in response to the wind $W2$), and the robot would fall over. With balance control the stability of the robot is maintained in spite of unexpected external forces.

the dynamic model of the robot inapplicable. Figure (7) shows the results when an external force $W1$ of $[0, 0, -5] N$ (a force from the right side) is applied at the torso of the robot during the interval $[5s, 10s]$, and another external force $W2$ of $[5, 0, -5] N$ (a force from the right and the back) is applied during the interval $[15s, 20s]$ (these two time

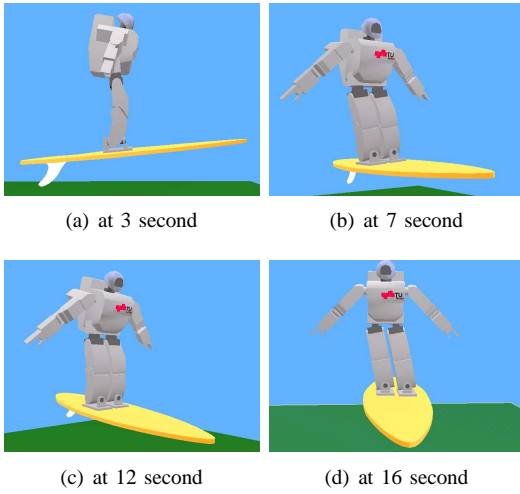


Fig. 8. Snapshots of the posture of the (simulated) HOAP-2 at 4 time points during the balancing experiment of Fig. 8. In (b) the wind $W1$ is blowing from the right side of the robot, and the robot is leaning against the wind in order to move its $pCoP$ back into the middle of the support polygon. In (d) another wind $W2$ is assumed to come diagonally from the right side and the back, and the robot also responds properly to this online change in its dynamic model.

intervals we shaded in gray). Note the onsets of the winds are abrupt and represent a highly dynamical perturbation to the system. The trajectory of the $pCoP$ is shown in Figure 7(c), 7(d) for the case of random movements of the platform in combination with external forces, both for the case with the described controller, and without. Without controller the robot lost balance after 16s (see black star in Figures 7(c), 7(e)), whereas with controller the balance is maintained. This was typical for the random movements of the platform and external forces that we have specified in the preceding text. Note that these external movements and forces were interpreted by the controller as external perturbations d (compare Figure 6), and were automatically counteracted. Resulting error signals which were fed back in the control loop for both dimensions x and z can be seen in Figures 7(e) and Figure 7(f) respectively.

In a third task we demonstrate that the setup as described can be used to follow any desired trajectory (i.e. time varying $\tilde{y}_{x,z}(t)$) in contrast to the preceding experiments where the robot has been controlled around a set point ($y_x = y_z = 0$). The robot stands now on flat ground. It should follow a desired trajectory of the shape of a figure "eight" (Figure 9(a)). The desired trajectory (gray;dash-dotted) splitted up in its corresponding dimensions x and z can be seen in Figures 9(b) and 9(c) respectively. Additionally the robot manipulates a heavy weight (20% of the robot's weight) with his left arm. The trajectory of the extended left arm over the simulation time of is shown in the first row of Figure 10. Note the arm joints are not under the regime of the KS 's but their movement change the $pCoP$. This can be clearly seen in an offset of the red dashed line in Figures 9(b) and 9(c) (when no control action is applied). For the control loop this is a disturbance d as any other external force and our proposed setup is able

to counteract it. Figures 9(b) (green line) and 9(c) (blue line) show that the robot is able to follow the desired trajectories despite the arm movement. The control loops react properly with sinusoidal movements of the KS -parameter to follow the desired trajectory and with an offsets to counteract the disturbance (see 9(d)). The second row of Figure 10 shows a series of screenshots. Note at the final position (last screenshot on the right) the robot leans to the right in order to get the $pCoP$ in the center of the support polygon.

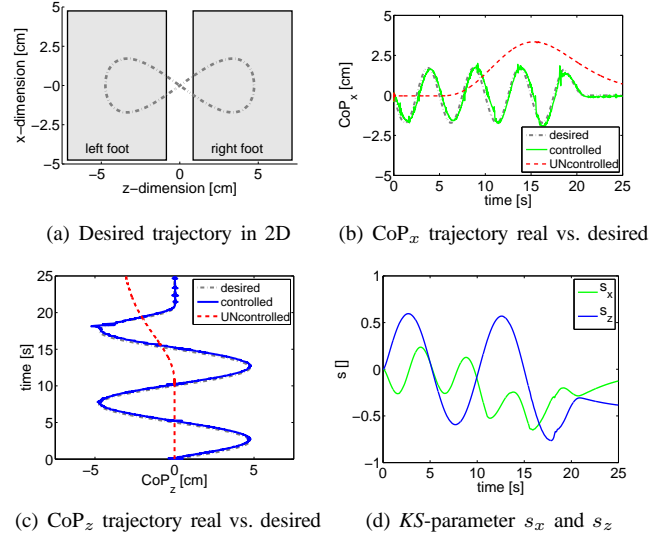


Fig. 9. Results of the trajectory following task: (a) presents the desired figure eight trajectory from bird view. The gray shaded areas depict the contact areas of the feet with the ground.(b) and (c) present the $pCoP$ trajectories splitted up into its two corresponding dimensions x and z respectively. Our approach is able to follow the desired trajectory (gray;dash-dotted). The red dashed lines depict the trajectory of the $pCoP$ if no control action is applied. (d) The system reacts properly with an offset in the KS -parameters to cope with the disturbances introduced by manipulating the weight.

VI. DISCUSSION AND GENERAL REMARKS

We demonstrated that biologically inspired kinematic synergies give rise to interesting new control methods for humanoid robots. We showed in experiments that our proposed setup is able to balance the humanoid robot HOAP-2 even under unknown external disturbances. The general strategy is to modularize and approximately linearize the control task for a robot with many degrees of freedom, so that simple controllers can be applied to the resulting virtual low-dimensional and approximately linear control problem. Linear control theory provides a large set of tools for improving the performance. One can also target the remaining nonlinearity and apply nonlinear control techniques [14].

The use of kinematic synergies of the type that we have discussed in this article may have an additional benefit for the control of humanoid robots, that needs to be explored in subsequent research: They make it feasible to apply powerful methods from machine learning such as reinforcement learning [15] to robot control, which typically require a low-

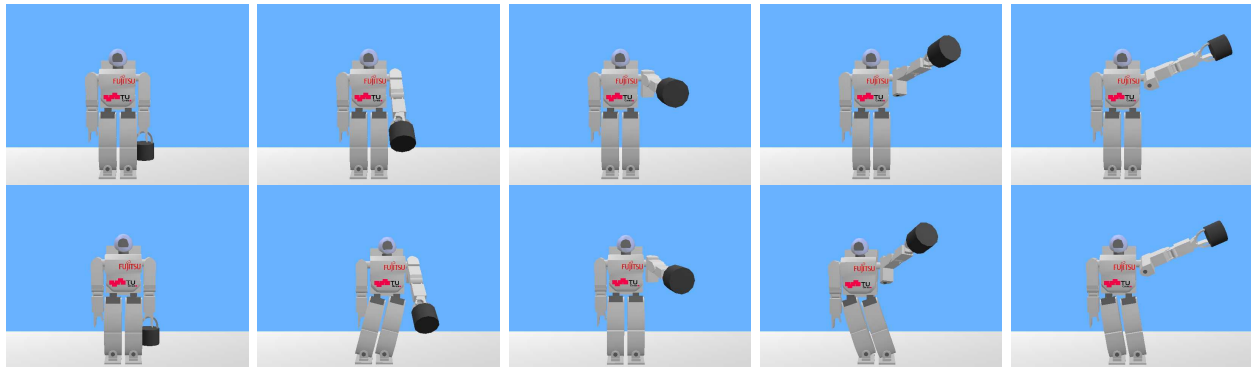


Fig. 10. Screenshots of the simulation for the trajectory following task: The corresponding times are from left to right 0 (start), 12.5, 15, 17.5, and 25 sec (end). In the first row no control action is applied. The plain arm trajectory of manipulating a heavy weight (20% of the robot's mass) is visible. The second row presents a series of screenshots when control is active. Note at the final position the robot leans to the right to cope with the heavy weight in his extended left arm. This results in a $pCoP$ at the center of the support polygon.

dimensional action space in order to escape the *curse of dimensionality*.

Another interesting new research direction suggested by the results of this article is the design of suitable kinematic synergies for a variety of movement tasks of humanoid robots. Besides the simple and straightforward design method that we have applied in this article, one can apply PCA to motion capture data of humans. In addition an interesting new research direction is the design of mathematical optimization methods for the construction of kinematic synergies with particular properties.

ACKNOWLEDGMENT

Written under partial support by the Austrian Science Fund FWF project # P17229.

REFERENCES

- [1] M. B. Popović, A. Goswami, and H. Herr, "Ground Reference Points in Legged Locomotion: Definitions, Biological Trajectories and Control Implications," *International Journal of Robotics Research*, vol. 24, no. 12, pp. 1013–1032, December 2005.
- [2] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba, and H. Inoue, "AutoBalancer: An Online Dynamic Balance Compensation Scheme for Humanoid Robots," in *Algorithmic and Computational Robotics: New Directions*, B. Donald, K. Lynch, and D. Rus, Eds. A.K. Peters Ltd., 2001, pp. 329–340.
- [3] J. J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Dynamically-stable Motion Planning for Humanoid Robots," *Autonomous Robots (special issue on Humanoid Robotics)*, vol. 12, pp. 105–118, January 2002.
- [4] F. A. Mussa-Ivaldi, "Modular features of motor control and learning," *Current Opinion in Neurobiology*, vol. 9, pp. 713–717, 1999.
- [5] A. d'Avella, P. Saltiel, and E. Bizzi, "Combinations of muscle synergies in the construction of a natural motor behavior," *Nature*, vol. 6, no. 3, pp. 300–308, March 2003.
- [6] A. d'Avella and E. Bizzi, "Shared and specific muscle synergies in natural motor behaviors," *Proc Natl Acad Sci*, vol. 102, no. 3, pp. 3076–3081, 2005.
- [7] S. M. S. F. Freitas, M. Duarte, and M. L. Latash, "Two kinematic synergies in voluntary whole-body movements during standing," *J Neurophysiol*, vol. 95, no. 2, pp. 636–645, Feb 2006.
- [8] V. Tricon, A. L. Pellec-Muller, N. Martin, S. Mesure, J.-P. Azulay, and S. Vernazza-Martin, "Balance control and adaptation of kinematic synergy in aging adults during forward trunk bending," *Neuroscience Letter*, vol. 415, no. 1, pp. 81–86, Mar 2007.

- [9] *HOAP2 Design Specification*. [Online]. Available: <http://jp.fujitsu.com/group/automation/downloads/en/services/humanoid-robot/hoap2/spec.pdf>
- [10] A. Goswami, "Postural stability of biped robots and the foot rotation indicator (FRI) point," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 523–533, 1999.
- [11] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*, 2nd ed. Springer, 1999.
- [12] A. V. Oppenheim and A. S. Willsky, *Signal and Systems*. Prentice-Hall Inc., Englewood Cliffs, 1992.
- [13] O. Michel, "Webots: Professional mobile robot simulation," *Journal of Advanced Robotics Systems*, vol. 1, pp. 39–42, 2004.
- [14] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, 1st ed. Prentice Hall, 1991.
- [15] R. S. Sutton and A. G. Barto, *Reinforcement Learning - An Introduction*, 1st ed. The MIT Press, 1998.