Sensors and Control Concept of Walking "Johnnie"

K. Löffler, M. Gienger and F. Pfeiffer

The International Journal of Robotics Research 2003; 22; 229
DOI: 10.1177/027836490302200307

K. Löffler
M. Gienger
F. Pfeiffer
Institute of Applied Mechanics
Technical University of Munich
Boltzmannstr. 15, 85747 Garching, Germany
loeffler@amm.mw.tu-muenchen.de

Sensors and Control Concept of Walking "Johnnie"

Abstract

One key problem to achieve a dynamically stable walking motion with biped robots is to measure and control the actual state of the robot with respect to its environment. Dynamically stable walking on unstructured terrain and fast walking can only be achieved with an orientation sensor. The control system of the biped robot "Johnnie" is designed such that the orientation of the upper body is controlled throughout all phases of the gait pattern. Furthermore, a sophisticated measurement and control of the foot torques has been implemented. In this way, the interaction forces and torques between robot and environment are controlled and tilting of the foot is avoided.

KEY WORDS—sensors, biped robot, control, dynamic stability, walking

1. Introduction

In the past few years, the development of biped walking robots has increased rapidly. The reasons for this are the fast improvements in the field of sensors, actuators and computers. Especially, the increase of computational power allows us to develop more sophisticated sensor fusion schemes and model-based control algorithms that lead to a stable and disturbance tolerant system behavior of such robots. Key developments have been achieved by Hirai, Hirose, and Takenaka (1998) and Kuffner et al. (2002) who have developed powerful biped walking robots.

The goal of this research project is the realization of a biped robot that is able to walk dynamically stable on even and uneven ground and around curves. It is also planned to realize a fast dynamically stable walking motion as well as "jogging" with flight phases. The planned gaits have been tested successfully in simulations. Currently, their implementation on the real robot is ongoing. The measured magnitudes are determined by joint position sensors, force sensors and an attitude reference system. The sensors are discussed in detail.

The control of biped walking robots is still a challenging problem. Different groups all over the world are working in this area, but still comprehensive solutions are not yet available. We have developed a general control concept for the biped walking robot "Johnnie", which has been tested in simulations and is now being implemented on a real robot.

Figure 1 shows the assembled robot "Johnnie". It is equipped with 17 joints. Each leg is driven with six joints: three in the hip, one in the knee and two (pitch and roll) in the ankle. The upper body has one degree of freedom (DOF) about the vertical axis of the pelvis. To compensate for the overall moment of momentum, each shoulder incorporates two DOF. The six DOF of each leg allow for an arbitrary control of the upper body's posture within the work range of the leg. Thus, the major characteristics of human gait can be realized. The geometry of the robot corresponds to that of a male human with a body height of 1.8 m (Hahn 1994). The total weight is about 40 kg. The biped is autonomous to a great extent; only power supply and currently part of the computational power are supplied by cables.

2. Sensor System

The sensor system can be subclassified in the internal joint sensors, in force sensors that measure the interaction with the environment and in an attitude sensor system that determines the upper body orientation of the robot with respect to the gravity vector.

2.1. Joint Sensors

Each joint is equipped with an incremental encoder (HP 5550) that is attached to the motor shaft. The encoders have two channels with 500 lines and one channel with a reference line. Such an accuracy of 1/2000 of a revolution is obtained using standard microcontroller hardware. The reference position for the measurement is obtained by light barriers. In an initialization phase, each joint passes the respective light barrier and uses the closest encoder reference line as a

The International Journal of Robotics Research
Vol. 22, No. 3–4, March–April 2003, pp. 229-239,
©2003 Sage Publications

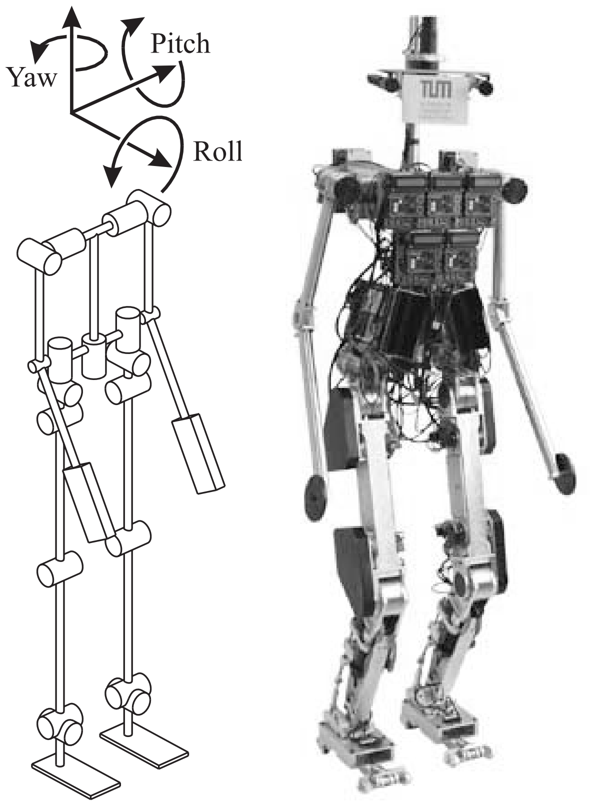

Fig. 1. “Johnnie”.

reference position. As the position is measured at the motor shaft, a high resolution is obtained.

In this procedure the elasticities of the gear components are neglected. However, as very stiff Harmonic Drive gears are employed, the structural elasticity does not lead to significant errors.

The joint angular velocity is determined by numerical differentiation of the position signal. Even though the resolution of the signal is low at high sampling rates, the overall accuracy compared to analog tachometers is high. Figure 2 shows a simulation of the measurement error of an analog tachometer compared to the differentiated encoder signal. The reference trajectory is a sinoidal velocity with a frequency of 1 Hz and an amplitude that is fitted to the 10 bit A/D conversion range. The sampling rate for the velocity calculation with the encoder is 3 ms. To reduce noise and ripple and to avoid digital errors such as aliasing, the tachometer signal is filtered with a 500 Hz PT2-filter. Figure 2 shows that the main error is due to the filter lag. The discretization of the encoder velocity signal leads to a higher noise level, but the amplitude of the error is lower.

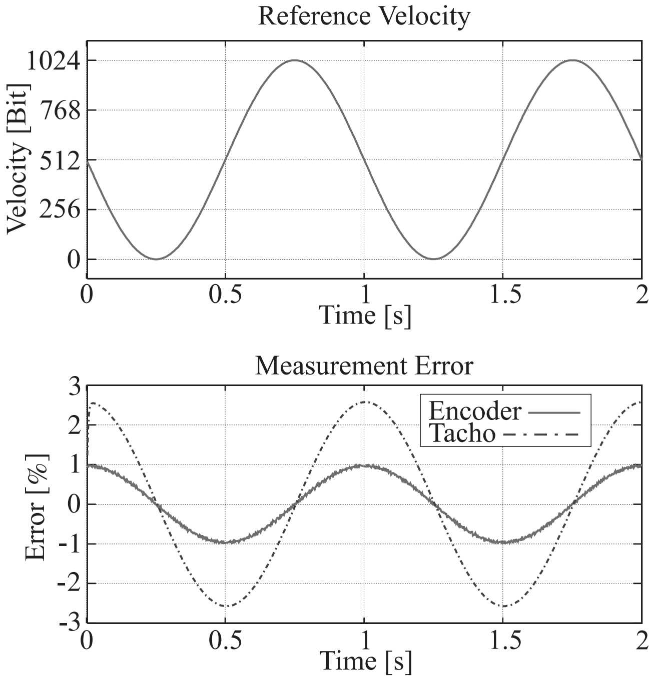

Fig. 2. Comparison of tachometer and encoder measurements.

2.2. Force Sensors

The interaction of the robot with its environment is confined to the feet. In the control of the forces, the ground geometry as well as the elastic and frictional properties have to be taken into account. The walking motion is mainly constrained by limitations of this contact situation. Exceeding force and torque limitations of the feet leads to slippage or tilt of the foot. Therefore, it is important to measure and control the ground reaction forces and torques. The biped robot “Johnnie” is equipped with two six-axes force/torque sensors that are integrated in the foot.

The geometry of the sensor has been developed on the basis of detailed simulations. The forces and torques acting on the foot for a jogging motion have been determined with a detailed multibody simulation program. Based on these data, a sensor layout has been chosen. Iterations of finite element analyses lead to the final design (Figure 3). The sensor consists of a single aluminum part. Three deformation beams holding strain gages are within the load path. Two opposing strain gages operate as a half bridge, thus compensating for temperature dependence. Thin membranes mechanically decouple the individual beam deflections to a great extent and so reduce cross talk. Special emphasis has been devoted to the strain gage application. The strain gages are selected to match the elastic properties of the sensor material. An exact application in combination with an appropriate temperature treatment finally leads to a high zero point stability of the signal.

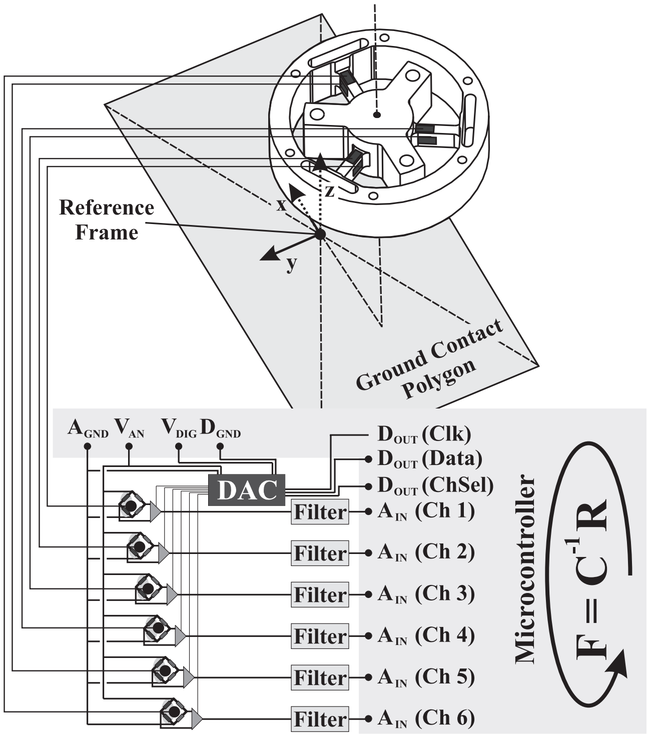

Fig. 3. Force sensor design.

The controller requires the force information with respect to a reference coordinate system (Figure 3). The six measured voltages are mapped onto the force components by matrix $\boldsymbol{C}^{-1}$ which results from a calibration procedure like that proposed, for example, in Galway (1980). The hardware realization can be seen in Figure 3. The bridge outputs are amplified to the A/D conversion voltage. In an initialization, the zero offset is computed and added to the bridge outputs by means of a D/A converter. The amplified voltages are filtered and then converted by a microcontroller (Infineon 167 CS). To reduce disturbances that result from the A/D conversion and the microcontroller hardware, further digital filtering of each channel is performed. With the resulting values, the reference force vector is computed, $\boldsymbol{F} = \boldsymbol{C}^{-1}\boldsymbol{R}$, where $\boldsymbol{F}$ denotes the six-dimensional vector of the reference force components, $\boldsymbol{R}$ denotes the six-dimensional vector of amplified bridge-voltages and $\boldsymbol{C}^{-1}$ is the projection matrix. This computation runs on the microcontroller as a real-time task with a sampling time of 0.5 ms.

2.3. Attitude Sensor

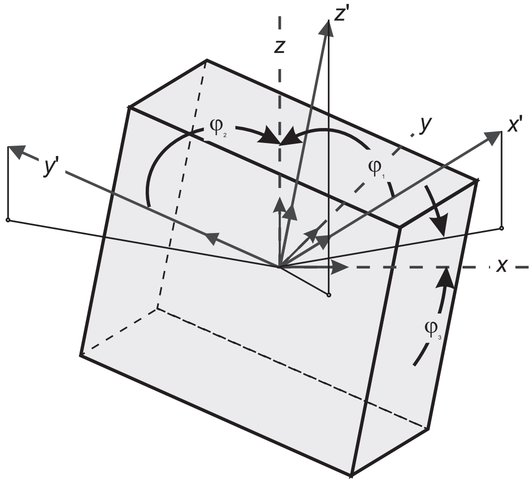

Fig. 4. Angular description of the upper body.

For a stable motion on uneven ground it is necessary to measure the orientation of the robot with respect to the gravity vector. Therefore, an attitude measurement system is integrated in the upper body. The angular description of the upper body has been proposed in Löffler, Gienger, and Pfeiffer (2000) and can be seen in Figure 4. The angles φ_1 and φ_2 denote the angles between the upper-body fixed x- and y-axis and the inertial z-axis. Angle φ_3 denotes the angle between the projection of the body-fixed x-axis on a horizontal plane and the inertial x-axis.

As the motion of the robot is associated with rather high accelerations in the walking cycle, the use of inclination sensors or accelerometers leads to poor results. The basis for the employed system is therefore the combination of gyroscopic sensors and accelerometers.

The angular velocities $\dot{\boldsymbol{\varphi}}$ are derived from the measured gyro-measurements $\boldsymbol{x}_{Gyro}$ with the inverse Jacobian $\boldsymbol{J}^{-1}$ that results from the geometry:

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{J}^{-1}\boldsymbol{x}_{Gyro}. \tag{1}$$

The respective angles φ_1 and φ_2 on the basis of the acceleration signals compute as

$$\varphi_1 = \arccos\left(\frac{a_x}{|\boldsymbol{g}|}\right) \tag{2}$$

$$\varphi_2 = \arccos\left(\frac{a_y}{|\boldsymbol{g}|}\right) \tag{3}$$

where $\boldsymbol{a}$ denotes the measured acceleration vector and $\boldsymbol{g}$ is the gravity vector.

Angle φ_3 cannot be measured by accelerations as it is parallel to the gravity vector. Instead, its angular velocity has to be integrated:

$$\varphi_3 = \int_0^t \dot{\varphi}_3(t')\mathrm{d}t'. \tag{4}$$

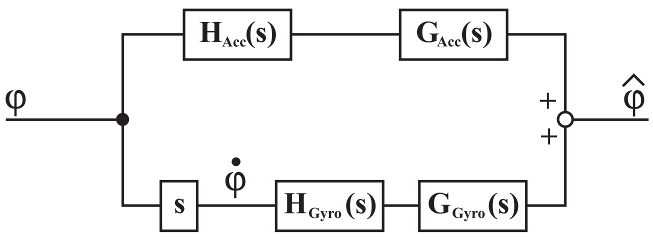

Fig. 5. Complementary filter.

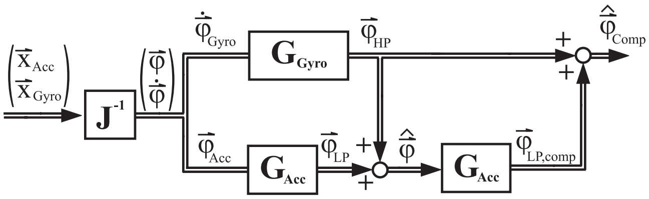

Fig. 6. Acceleration compensation.

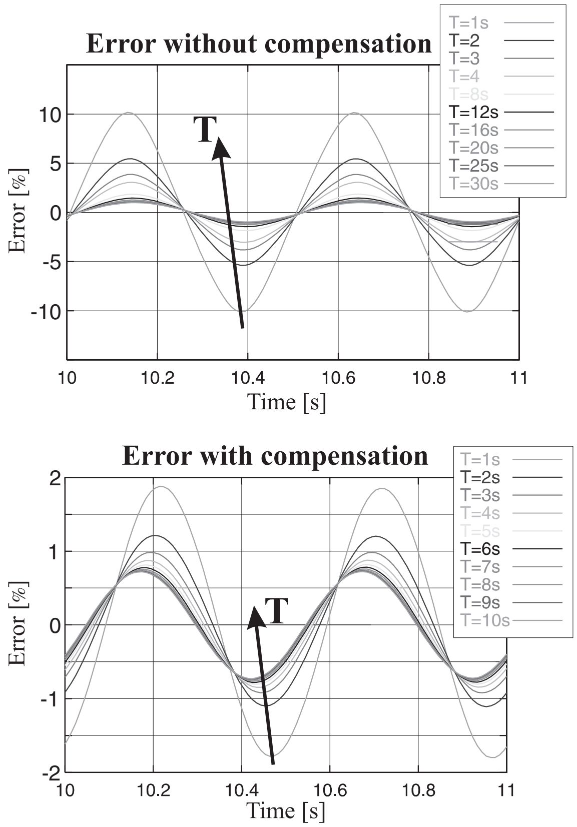

Fig. 7. Acceleration compensation.

However, problems arise when the orientation is determined by the gyros and accelerometers separately. The acceleration sensors produce erroneous results when the body rotation is superimposed by translational accelerations. On the other hand, integration of the velocity data leads to an unbounded error due to noise and disturbances in the measured gyro signals. To overcome these problems, a variety of sensor fusion methods has been proposed (Foellinger 1992; Maybeck 1979). The scheme that is employed here is based on complementary filtering of the gyro- and accelerometer-based signals. The basic idea is to weight the sensor data in frequency ranges where the respective sensor can be considered as ideal. Figure 5 shows a block diagram of the filter. H_{Acc} and H_{Gyro} denote the sensor transfer functions, and G_{Acc} and G_{Gyro} comprise the respective filter functions of the sensor.

The condition that the estimate equals the real orientation leads to an infinite number of possible filter functions. As the filter functions are chosen to match the properties of the respective sensor, the sensor transfer functions may be considered as $H_{Acc} = H_{Gyro} = 1$. As proposed in Baerveldt and Klang (1997), the velocity is integrated and high-pass filtered. This has the advantage that constant drift is compensated. The orientation computed by the acceleration sensors is low-pass filtered. The time constant of low- and high-pass filters is equal and defines the estimation behavior.

Choosing a low time constant leads to weighting the accelerations high, thus making the estimate become sensitive to linear accelerations. Choosing a high time constant is advantageous as it takes advantage of the good dynamic properties of the gyroscopes. On the other hand, the time-varying bias is not compensated so fast and emerging disturbances decay slower.

One shortcoming of this method is that the low-pass branch leads to results that are corrupted by translational accelerations. At higher frequencies, their portion is smaller. To overcome this problem, the superimposed acceleration is compensated using the orientation estimate (Figure 6).

This first orientation estimate is already very close to the real orientation and serves therefore as input for the low-pass filter. This corrected low-pass orientation is then added to the high-pass orientation, thus leading to a very accurate result. The dynamic properties of the corrected estimate are merely confined by the response of the gyroscopes. The compensation allows for lower filter time constants such that a fast error decay is realized. Figure 7 shows the simulated estimation errors with and without compensation. The excitation is a sinoidal orientation trajectory with a frequency of 1 Hz and an amplitude of 90°.

Figure 8 shows a CAD drawing of the realized sensor. To reduce electromagnetic disturbances, all sensors and the microcontroller are integrated in the sensor housing. The interface to the sensor is the supply voltage and the bus interfaces (CAN, serial interface). The filter runs with a sampling time of 2 ms. Table 1 shows the important sensor data.

Table 1. Orientation Sensor

Gyros	Silicon Sensing Systems	CRS 04
	Bandwidth	85 Hz
	Range	$+/-150$ deg s^{-1}
Accelerometer	Crossbow	CXL02TG3
	Range	$+/-2$ g

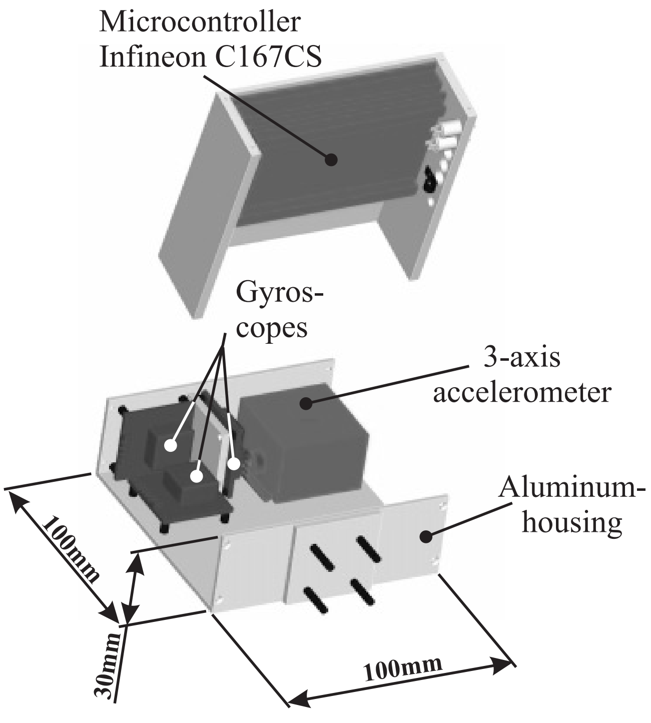

Fig. 8. Realized orientation sensor.

3. Control

3.1. Constraints

The main difficulties in the control of dynamically walking robots result from constraints that limit the applicability of conventional control concepts. Two groups of constraints need to be considered. Firstly, the workspaces of the joints (5), the maximum rotor velocities (6) and the joint torques (7) are limited:

$$\boldsymbol{q}_{min} \le \quad \boldsymbol{q} \quad \le \boldsymbol{q}_{max} \tag{5}$$

$$\dot{\boldsymbol{q}}_{min} \le \quad \dot{\boldsymbol{q}} \quad \le \dot{\boldsymbol{q}}_{max} \tag{6}$$

$$\boldsymbol{\lambda}_{min}(\dot{\boldsymbol{q}}, \boldsymbol{T}) \le \quad \boldsymbol{\lambda} \quad \le \boldsymbol{\lambda}_{max}(\dot{\boldsymbol{q}}, \boldsymbol{T}). \tag{7}$$

These are typical constraints for industrial robots and can be satisfied by an adequate design and an appropriate choice of the trajectories. However, critical control problems result from the second group of constraints that describe the unilateral contact between the feet and the ground. Depending on the normal force $F_{i,z}$ that is transmitted from the foot $i = 1, 2$ to the ground, the maximum transmissible torques $T_{i,x}$, $T_{i,y}$ and $T_{i,z}$, as well as the tangential forces $F_{i,x}$ and $F_{i,y}$ are limited by the size of the feet l_x, l_y and the coefficients of friction μ_t, μ_d:

$$|T_x| \le 0.5\ F_z l_y \quad |T_y| \le 0.5\ F_z l_x \tag{8}$$

$$\sqrt{F_x^2 + F_y^2} \le \mu_t F_z \qquad F_z \ge 0 \quad |T_z| \le \mu_d F_z. \tag{9}$$

While practical experiments show that the robot usually does not start slipping, the limits of the torques in the lateral and frontal directions T_x and T_y lead to a small margin of stability. Much research has been carried out on concepts to ensure that these constraints are satisfied throughout the entire gait cycle. The "zero moment point" (ZMP) theory is one of the most popular approaches to describe the constraints (Vucobratovic et al. 1990).

3.2. Trajectory Generation

The trajectories of the robot are defined in terms of Cartesian coordinates. These are the position of the center of gravity $\boldsymbol{x}_{cog}$, the rotation of the upper body $\boldsymbol{\varphi}_U$ and the position $\boldsymbol{x}_{Fs}$ and orientation $\boldsymbol{\varphi}_{Fs}$ of the foot that is swinging forward. Additionally, the pelvis joint q_p and the joints of the arms $\boldsymbol{q}_a$ are controlled. Then $\boldsymbol{x}_{ref} = (\boldsymbol{x}_{cog}^T\ \boldsymbol{\varphi}_U^T\ \boldsymbol{x}_{Fs}^T \boldsymbol{\varphi}_{Fs}^T\ q_p\ \boldsymbol{q}_a^T)^T$ is the vector of controlled variables. Except for the horizontal motion of the center of gravity, the motion of these variables is defined in fifth-order polynomials for each phase of the gait pattern.

The reference trajectories of the center of gravity are computed with a lumped mass model. These approximations are not completely exact, as the acceleration of the swinging foot does also influence the dynamics of the center of gravity. However, practical experiments have shown that the model is sufficient for walking speeds up to 2.0 km h^{-1}.

In the reduced model, it is assumed that the mass of the robot can be lumped to the center of gravity. The motion of the center of gravity in the frontal direction is independent of the motion in the lateral direction. When the center of gravity is kept at a constant height, we obtain a particularly simple solution for the dynamics of the center of gravity. For the lateral direction y_{cog} the acceleration $\ddot{y}_{cog}$ results as

$$\ddot{y}_{cog} = \frac{g_z}{z_{cog}}(y_{cog} - y_{zmp}). \tag{10}$$

Here g_z is the vertical component of the gravity vector, z_{cog} is the height of the center of gravity and y_{zmp} is the position of the ZMP. During walking, the center of gravity is shifted periodically from one leg to the other such that the legs can alternately swing forward. During the single support phase, the lateral position of the ZMP point is constant with respect to the supporting foot. For maximum stability margins it can

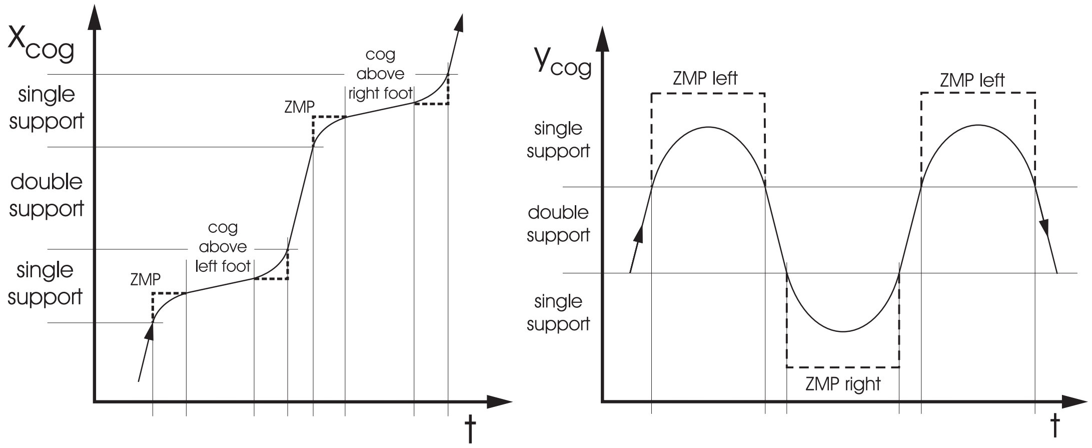

Fig. 9. COG frontal/lateral.

be selected to be in the middle of the foot area; for minimum lateral deviation of the center of gravity it has to be on the inner edge of the supporting foot. The resulting equations for single support are

$$y_{cog} = c_1\cosh(a(t - t_0)) + c_2\sinh(a(t - t_0)) \tag{11}$$

with $a = \sqrt{\frac{g_z}{z_{cog}}}$. The coefficients c_1 and c_2 are computed such that $y_{cog}(t_0) = y_{cog}(t_1)$ and $\dot{y}_{cog}(t_0) = -\dot{y}_{cog}(t_1)$ with t_0 and t_1 being the beginning and the end of the single support phase. During double support the velocity of the center of gravity is constant. The resulting motion of the center of gravity is depicted in Figure 9 (right).

The velocity of the center of gravity in the walking direction is computed according to the same principle. During the single support phase, the ZMP moves from the rear edge of the supporting foot to the front edge. In this way, the velocity of the center of gravity can be kept constant while it is above the supporting foot. The corresponding motion of the center of gravity in the frontal direction is depicted in Figure 9 (left). The simplified model has the advantage that the trajectories can be computed online and therefore it is possible to compensate model inaccuracies as well as external disturbances by an adaptation of the trajectories.

3.3. Computed Torque Method

The computed torque method allows us to consider the entire system dynamics in the control of the robot. The dynamics of the system is denoted:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{W}_1 \begin{pmatrix} \boldsymbol{\lambda}_m \\ T_x \\ T_y \end{pmatrix} = \boldsymbol{h} - \boldsymbol{W}_F \boldsymbol{\lambda}_{FR}. \tag{12}$$

Here $\boldsymbol{M} \in \mathbb{R}^{21\times 21}$ is the mass matrix of the entire system, $\boldsymbol{q} \in \mathbb{R}^{21}$ are the generalized coordinates and $\boldsymbol{h} \in \mathbb{R}^{21}$ is the vector of nonlinear dynamic terms including the Coriolis forces. The forces and torques that act on the system are split up in four terms. T_x and T_y are the lateral and frontal torques between the supporting foot and the ground. These are a function of the torques of the corresponding ankle joint, therefore the torques of the ankle joint do not explicitly show up in the equations. Vector $\boldsymbol{\lambda}_m$ comprises the remaining 15 joint torques. The forces that act on the supporting foot and the torque around the vertical axis are contained in $\boldsymbol{\lambda}_{FR}$. Using the Jacobians $\boldsymbol{W}_1$ and $\boldsymbol{W}_F$, the torques and forces are mapped on the generalized coordinates.

As described above, the trajectories are computed in terms of Cartesian coordinates $\boldsymbol{x}_{ref} = (x_{cog} y_{cog} z_{cog} \boldsymbol{\varphi}_U^T \boldsymbol{x}_{Fs}^T \boldsymbol{\varphi}_{Fs}^T q_p \boldsymbol{q}_a^T)^T \in \mathbb{R}^{17}$. As long as the system is not underactuated, we can define a linear behavior for these variables:

$$\ddot{\boldsymbol{x}} = \ddot{\boldsymbol{x}}_{ref} + \boldsymbol{A}_1\Delta\dot{\boldsymbol{x}} + \boldsymbol{A}_2\Delta\boldsymbol{x} + \boldsymbol{A}_3\int\Delta\boldsymbol{x}\,\mathrm{d}t \tag{13}$$

$$\text{with} \quad \Delta\boldsymbol{x} = (\boldsymbol{x}_{ref} - \boldsymbol{x}). \tag{14}$$

The Cartesian coordinates are mapped on the generalized coordinates with Jacobian $\boldsymbol{W}_2$, such that

$$\boldsymbol{W}_2\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{x}}_{ref} + \boldsymbol{A}_1\Delta\dot{\boldsymbol{x}} + \boldsymbol{A}_2\Delta\boldsymbol{x} + \boldsymbol{A}_3\int\Delta\boldsymbol{x}\,\mathrm{d}t - \dot{\boldsymbol{W}}_2\dot{\boldsymbol{q}}. \tag{15}$$

In the following, the control law is summed up in $\boldsymbol{w}$:

$$\boldsymbol{w} = -(\ddot{\boldsymbol{x}}_{ref} + \boldsymbol{A}_1\Delta\dot{\boldsymbol{x}} + \boldsymbol{A}_2\Delta\boldsymbol{x} + \boldsymbol{A}_3\int\Delta\boldsymbol{x}\,\mathrm{d}t - \dot{\boldsymbol{W}}_2\dot{\boldsymbol{q}}. \tag{16}$$

The motor torques and the foot torques are computed with eqs. (15) and (16) together with the equations of motion (12):

$$\begin{pmatrix} \boldsymbol{\lambda}_m \\ T_x \\ T_y \end{pmatrix} = (\boldsymbol{W}_2\boldsymbol{M}^{-1}\boldsymbol{W}_1)^{-1}(\boldsymbol{w} + \boldsymbol{W}_2\boldsymbol{M}^{-1}(\boldsymbol{h} - \boldsymbol{W}_F\boldsymbol{\lambda}_{FR})). \tag{17}$$

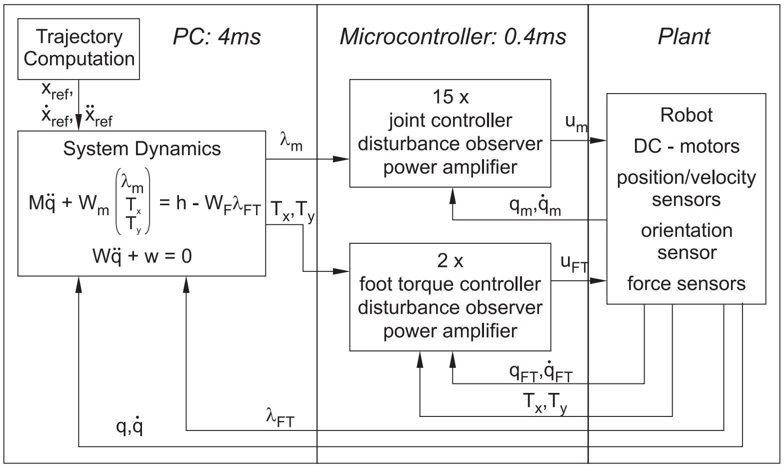
Trajectory Computation
PC: 4ms
Microcontroller: 0.4ms
Plant
x_{ref}, $\dot{x}_{ref}$, $\ddot{x}_{ref}$
System Dynamics
$M\ddot{q} + W_m \begin{pmatrix} \lambda_m \\ T_x \\ T_y \end{pmatrix} = h - W_F\lambda_{FT}$
$W\ddot{q} + w = 0$
λ_m
T_x, T_y
15 x joint controller disturbance observer power amplifier
2 x foot torque controller disturbance observer power amplifier
u_m
u_{FT}
Robot
DC - motors
position/velocity sensors
orientation sensor
force sensors
$q_m, \dot{q}_m$
$q_{FT}, \dot{q}_{FT}$
T_x, T_y
λ_{FT}
$q, \dot{q}$

Fig. 10. Computed torque.

3.3.1. Torque Constraints

The resulting torques lead to the desired system behavior, as long as the limits of T_x and T_y are not exceeded. When the result of eq. (17) violates the constraints from eq. (8), a different set of controlled variables is chosen. For example, when the limits of T_x are exceeded, we give up controlling the horizontal position of the center of gravity in the lateral direction y_{cog}. Then the vector of controlled variables has only 16 elements: $\boldsymbol{x}^*_{ref} = (y_{cog}\ z_{cog}\boldsymbol{\varphi}_U^T\boldsymbol{x}_{Fs}^T\ \boldsymbol{\varphi}_{Fs}^T\ q_p\ \boldsymbol{q}_a^T)^T \in \mathbb{R}^{16}$. Instead of the horizontal position of the center of gravity we can now control the foot torque T_x, which is equal to the extremal value ($T_x = T_{x,min}$ or $T_x = T_{x,max}$, respectively). In the resulting equations of motion, the limited torque T_x is treated like an external force, that is mapped on the equations of motion with the Jacobian $\boldsymbol{W}_{T,x}$:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{W}_1^* \begin{pmatrix} \boldsymbol{\lambda}_m \\ T_y \end{pmatrix} = \boldsymbol{h} - \boldsymbol{W}_F\boldsymbol{\lambda}_{FR} - \boldsymbol{W}_{T,x}T_x. \qquad (18)$$

Considering that the mapping of the controlled variables has to be adapted to the reduced set, we introduce the reduced Jacobian $\boldsymbol{W}_2^*$ and the reduced right-hand side $\boldsymbol{w}^*$. Then the joint torques and the foot torque T_y are computed:

$$\begin{pmatrix} \boldsymbol{\lambda}_m \\ T_y \end{pmatrix} = (\boldsymbol{W}_2^*\boldsymbol{M}^{-1}\boldsymbol{W}_1^*)^{-1}(\boldsymbol{w}^* + \boldsymbol{W}_2^*\boldsymbol{M}^{-1}(\boldsymbol{h} \\ -\boldsymbol{W}_F\boldsymbol{\lambda}_{FR} - \boldsymbol{W}_{T,x}T_x)). \qquad (19)$$

The torque in the frontal direction T_y is treated in the same way, when it reaches its maximum/minimum value. In this case we give up controlling the horizontal position of the center of gravity in the frontal direction.

In the practical implementation it is not necessary to compute both eqs. (17) and (19) and all other possible combinations of limited minimal and maximal values of the foot torques. In order to find a valid solution for the torque distribution, it is more efficient to transform the problem into a linear complementarity problem (LCP), which can be solved with less computational effort.

Based on the presented scheme, the motor torques are determined consistently for a given limitation of the foot torques. In this way it is ensured that the robot does not tip over in case of disturbances, but it will just accelerate horizontally, which is not critical for the system stability.

The concept to accelerate/decelerate the robot when the upper body tilts forward/backward is known from other papers (Hirai, Hirose, and Takenaka 1998). However, up to now no consistent solution to determine the system dynamics has been presented for the underactuated case. Using the mapping concept, the exact solution results directly from the set of controlled variables.

The control scheme has been implemented on our biped robot "Johnnie" to realize a dynamically stable walking motion. As shown in Figure 10, the computation of the reference trajectories and the computation of the system dynamics is performed on a PC running under a real-time LINUX kernel (RTAI). The computed torques are sent to decentral microcontrollers (Infineon C167CS) that drive the power amplifiers and read in the sensor data.

Experiments show that the the actual torques of the joints do significantly depend on the time variant friction of the gears. Therefore, disturbance observers are used to estimate external disturbances as well as varying motor parameters and friction.

As discussed in Section 3.1, the foot torques are of particular importance for the system stability. While the actual torques of the joints can only be estimated, it is possible to use the information of the foot torque sensor for the control of the ankle joint.

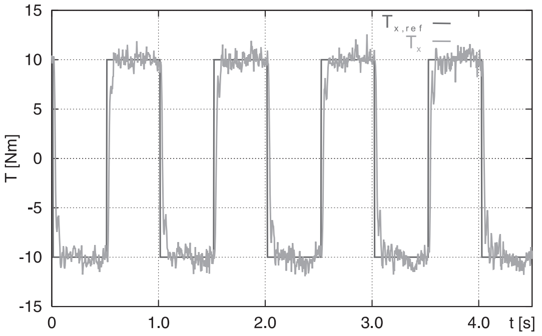

Fig. 11. Torque control.

In our hardware implementation, the microcontroller that drives the ankle joint is also used to evaluate the force sensor. In this way, it is possible to realize a sampling time of 0.4 ms for the torque control of the feet. Experiments have shown that the performance of the torque controller can be improved by a disturbance observer. The performance of the controller is shown in Figure 11 for a rectangular reference input.

3.4. Problems of the Computed Torque Method

Theoretically, the presented scheme leads to an optimal system performance, since all dynamical effects and the limitation of the foot torques are considered.

The performance of the controller has been investigated with the real robot in practical experiments. A walking speed of 1.2 km h^{-1} can be achieved with step lengths of 0.35 m. Even though this is in the range of present state walking machines (Honda Asimo), it has become obvious that the control scheme cannot be used for higher walking speeds. In spite of a fast control of the ZMP, the overall system bandwidth is very low. This is due to three problems, as follows.

- Due to the high computational effort, it is not possible to compute the overall system dynamics in less than 4 ms on a Pentium III processor 800 MHz.
- The communication between microcontrollers and PC requires more than 1 ms with the employed CAN-bus system.
- The orientation sensor has a crossover frequency of 85 Hz, leading to a considerable time delay of the orientation velocity signal.

The time delay due to the computation time of the PC can be reduced with more efficient algorithms and faster processors, and the communication time can be reduced with a bus system with higher bandwidth. However, it is not possible to obtain an orientation sensor with a significantly higher crossover frequency. Therefore, it cannot be expected that faster gait patterns can be realized with this control scheme.

3.5. Joint Position Control

In order to reach higher walking velocities, a different control approach has been investigated, which is based on typical implementations that are used for industrial robots. A central trajectory generator computes the reference position and velocity of each joint and sends these values to the joint controllers. Even though the dynamic effects are not considered, a very good trajectory tracking can be achieved due to a high sampling rate of the joint position controllers. While this approach is efficient for fully actuated robots, it is not obvious that the method can be implemented efficiently on an underactuated robot. In particular, the control of the foot torques requires a fast adaptation of the trajectories.

For our robot, the control scheme has been implemented according to Figure 12. Again the reference trajectories are computed in terms of Cartesian coordinates. For the computed torque method, it was sufficient to determine the Jacobian that maps the velocities of the controlled variables on the velocities of the generalized coordinates. Now the kinematic transformation from $\boldsymbol{x}$ to $\boldsymbol{q}$ has to be computed. We use a Newton iteration to compute the transformation. The computational effort is low compared to the effort for the computation of the system dynamics. The joint positions and velocities are controlled by the microcontrollers with a sampling rate of 0.4 ms.

The overall system stability is now investigated with a linearized model. Mass and inertia of the robot are summed together such that we obtain an inverted pendulum model according to Figure 13, as often used for the computation of the reference trajectories.

For a real robot, the contact to the ground has always some compliance. This results from the stiffness of the links, the elasticity of the foot elements and the compliance of the ground surface. In the model, all of these elasticities are combined to a contact stiffness C_F. Typically, the damping of the contact is relatively low. When the motion of the joints is independent of the orientation of the upper body, the system is obviously marginally stable. This is also the case for an ideally stiff system, when $C_F \to \infty$. The undamped system dynamics leads to instability within a few steps. As soon as one foot hits the ground at the wrong time, the robot tips over.

Therefore, it is necessary to adapt the trajectories of the robot in a suitable way. In our control scheme, the trajectories of the ankle joint are modified depending on the orientation of the upper body and the torques between foot and ground. As depicted in Figure 12, the control architecture is similar to an impedance control.

The stability of the control scheme is demonstrated here for the simplified model. Figure 14 shows a block diagram for the reduced system. The dynamics of the actuator is modeled as a second-order system, and the motion of the upper body φ_O results from the torque of the contact stiffness. With the microcontrollers we can control the angle of the foot φ_F rela-

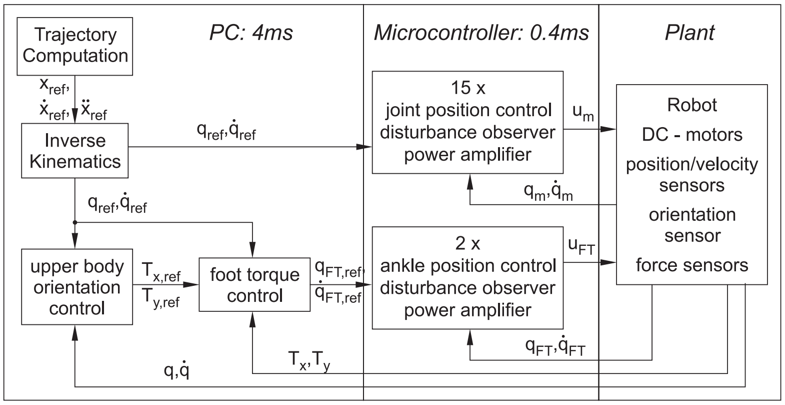

Fig. 12. Trajectory control.

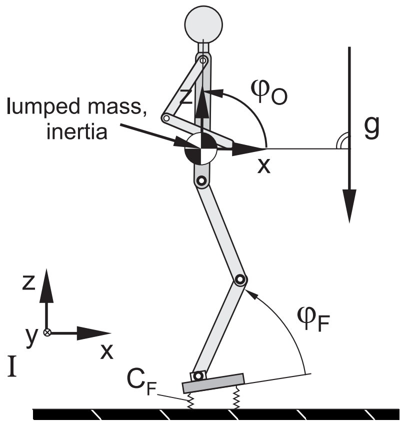

Fig. 13. Inverted pendulum.

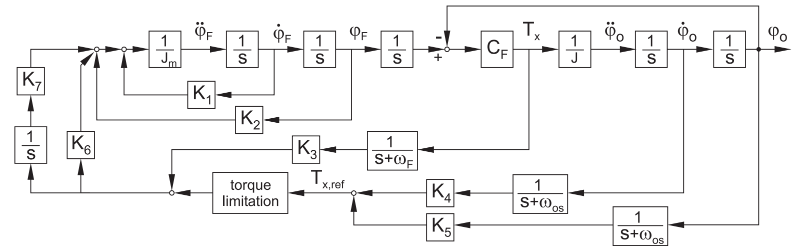

Fig. 14. Control of inverted pendulum model.

tive to the robot (gains K_1 and K_2). As well as the position and velocity of the ankle joint, the contact torques (gain K_3) and the orientation (gain K_5) and rotational velocity (gain K_4) of the upper body are also fed back. The transfer function of the orientation sensor is approximated by a PT-1 behavior, and the force sensors are filtered with a crossover frequency of 250 Hz.

Figure 14 shows that the poles for the closed-loop system can be placed to achieve a high system bandwidth. For the linearized model, a linear controller with appropriate feedback gains $K_1 \ldots K_7$ leads to an optimal system behavior.

4. Conclusions

The control of a biped robot requires a comprehensive set of sensors to determine the position of the robot in space and the forces and torques that act on the system. In this paper, we present the biped robot "Johnnie". In particular, we discuss the joint position sensors, the design of a three-dimensional orientation sensor and the six-axes force/torques sensor.

Two different control concepts are presented. First an implementation based on the method of feedback linearization is discussed. Considering the entire system dynamics, it allows us to compute the torques of each joint such that the robot follows a given trajectory. Even though the method is theoretically correct, it is not the optimal solution for a real robot, as the computational effort is very high and it is necessary to have very accurate sensors with a high bandwidth.

The second control concept is similar to the typical implementation of control systems for industrial robots. The trajectories are computed on a central PC and sent to independent joint controllers that operate at a high sampling rate. In order to ensure the stability of the upper-body orientation, a foot torque controller adapts the trajectories of the ankle joints such that the robot does not tilt considering the limits of the foot torques.

The presented control scheme has been verified in experiments. At present, stable walking can be realized with up to 2.0 km h^{-1} and step lengths of 55 cm. Figure 15 shows the robot walking on a conveyor belt. With an external PC, the operator can send control commands to the robot. The parametric definition of the trajectories allows us to change the step length/walking speed and the height and orientation of the upper body during walking. The trajectories are adapted online such that a dynamically stable motion results. In the same way, the operator can prescribe a certain turning angle for each step, such that the robot can walk around curves.

Additionally, the robot can step over obstacles and climb up stairs with a height of 12.5 cm per stair. The ultimate goal of this project is to achieve a jogging motion. Currently trajectories are developed for higher walking speeds.

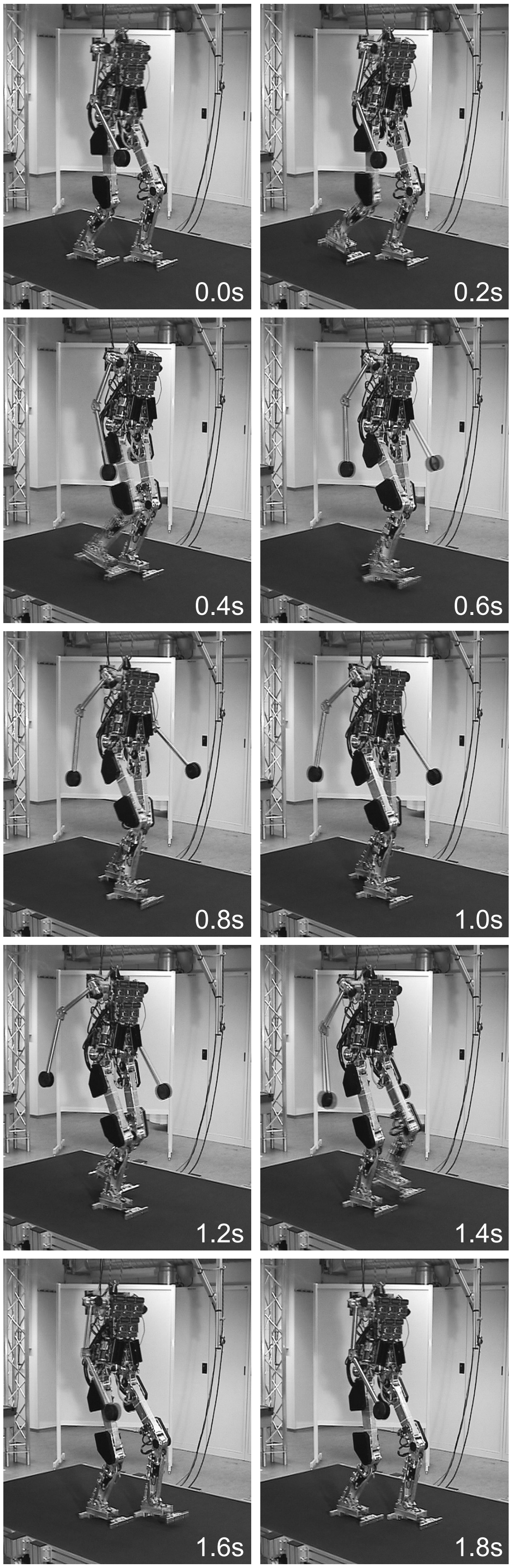

Fig. 15. "Johnnie" walking.

Acknowledgment

This project has been supported by the “Deutsche Forschungsgemeinschaft” within the priority program “Autonomous Walking”.

References

Baerveldt, A.-J., and Klang, R. 1997. A low-cost and low-weight attitude estimation system for an autonomous helicopter. In *Proc. IEEE Int. Conf. on Intelligent Engineering Systems*, Budapest, Hungary, pp. 391–395.

Foellinger, O. 1992. *Regelungstechnik.* Hüthig Buch Verlag, Heidelberg, Germany.

Galway, R. D. 1980. A comparison of methods for calibration and use of multi-component strain gauge wind tunnel balances. Report LR-600, National Research Council, Ottawa, Canada.

Hahn, U. 1994. Calculation of anthropometric data for human body segments. Implemented in Software Program “Calcman3d”.

Hirai, K., Hirose, M., and Takenaka, T. 1998. The development of Honda humanoid robot. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, pp. 160–165.

Gienger, M., Löffler K., and Pfeiffer, F. 2001. Towards the design of a biped jogging robot. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, pp. 4140–4145.

Löffler, K., Gienger, M., and Pfeiffer, F. 2000. Control of a biped jogging robot. In *Proc. 6th Int. Workshop on Advanced Motion Control*, Nagoya, Japan, pp. 307–323.

Kuffner, J. J., Kagami, S., Inoue, H., Inaba, M., and Nishiwaki, K. 2002. Dynamically stable motion planning for humanoid robots. *Autonomous Robots* 12:105–118.

Maybeck, P. S. 1979. *Stochastic Models, Estimation and Control*, Academic, New York.

Pfeiffer, F., Eltze, J., and Weidemann, H.-J. 1995. The TUM-walking machine. In *Intelligent Automation and Soft Computing*, pp. 307–323.

Pfeiffer, F., Löffler, K., and Gienger, M. 2002. The concept of jogging Johnnie. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington DC, USA, pp. 3129–3135.

Pfeiffer, F., and Steuer, J. 1999. Design of walking machines—control aspects. *IFAC, 14th Triennial World Congress*, Bejing, China, pp. 413–418.

Rossmann, Th., and Pfeiffer, F. 1996. Control and design of a pipe crawling robot. In *Proc. 13th IFAC World Congress of Automatic Control*, San Francisco, USA.

Vucobratovic, J., Borovac, B., Surla, D., and Stokic, D. 1990. *Biped locomotion: dynamics, stability, control and applications*, Springer-Verlag, Berlin.

Weidemann, H.-J. 1993. *Dynamik und regelung von sechsbeinigen robotern und natürlichen hexapoden.* Fortschritt-Berichte VDI, Reihe 8, Nr. 362, VDI-Verlag, Düsseldorf.