Virtual Pivot Point Control for Running Robots

Delft Biorobotics Laboratory, Mechanical Engineering, Delft University of Technology
Leonard van Bommel, Advisor: Daniël Karssen
September 5, 2011

Abstract—Keeping the torso of a running robot upright is a complex task, as the torso is subject to a multitude of forces throughout the running cycle. Often a PD-control scheme is used to keep the torso in the desired upright orientation. Recently, a new torso control scheme was introduced, called Virtual Pivot Point (VPP) control. The VPP-controller stabilises the torso by applying a hip torque such that the Ground Reaction Force (GRF) is directed at a fixed point on the torso. The objective of this simulation study is to compare the performance of the VPP-controller against the PD-controller. Performance is quantified by the maximum allowable disturbance for which the simulation model can continue running. Computationally demanding simulations are performed on a simple running model and accurate results are acquired on a realistic running model. Results show that PD control outperforms VPP control in all tested situations. In theory VPP control is a promising, but its implementation needs improvement.

Index Terms—running robot, torso, CoM offset, Virtual Pivot Point, disturbance rejection.

I. INTRODUCTION

Bipedal running is a complex task, as it is a highly dynamical gait that involves a flight phase, during which no force can be applied to the environment. Humans are able to endure large disturbances during their running gait, while this performance is still unprecedented in robotic running. However, studying humanlike running gaits on robots can result in a fundamental understanding of the principals governing stable running. This research has two major benefits. Firstly, legged locomotion creates highly mobile robots, that could be used to navigate through rough terrain and reach locations that would be inaccessible to wheeled robots. Furthermore, knowledge gained when studying legged locomotion, and in particular bipedal locomotion, can be used to devise new rehabilitation techniques or it can aid in the improvement of existing prostheses for leg amputees.

Additionally, a running bipedal gait allows for forward movement greater than the maximum walking speed, as the walking speed is upper bound by \( \sqrt{g} \) \(^1\) [1], [2]. Summarising, bipedal running allows for movement at greater speeds while maintaining maximum mobility.

A challenging part of bipedal running is to keep the torso in upright position, as the torso resembles an unstable inverted pendulum. Cameras and other sensory systems in the torso can be disturbed by large fluctuations of the torso orientation. Additionally, the movement of the torso can disturb the overall running dynamics. The most common way to stabilise the torso is to use a PD-controller that applies a torso torque based on differences between the current and the desired torso pitch and

\[ T_{\text{torso}} \]

on the torso pitch angular velocity. A large number of running robots have shown that such a simple PD-controller can indeed stabilise the torso during a running motion [3]–[13].

The use of PD-control for stabilising the torso might not be the optimal control solution as the control output is invariant to the orientation of the leg, which is counterintuitive. Recently, Maus et al. introduced a new control scheme, called Virtual Pivot Point control [14], [15], which does take the leg orientation into account. This controller stabilises the torso by applying a torso torque such that the Ground Reaction Force (GRF) is directed at a fixed point on the torso, called the Virtual Pivot Point (VPP). When the GRF always directs at the VPP, the VPP becomes a virtual hinge around which the torso will rotate. Essentially, this transforms the difficult task of balancing an inverted pendulum to a system consisting

\[ \text{Figure 1: Schematic working of the Virtual Pivot Point (VPP) controller for three different torso pitch angles.} \]

\[ \text{Left: the VPP is aligned with the leg, therefore no torso torque is applied.} \]

\[ \text{Center: the torso is tilted backwards with respect to the leg, therefore no torso torque is applied.} \]

\[ \text{Right: the torso is tilted forwards with respect to the leg, therefore a positive torso torque is used to direct the GRF at the VPP.} \]

\[ \text{Footnotes:} \]

\[ ^{1} \text{where} \ l \ \text{is the typical leg length and} \ g \ \text{is the gravitational acceleration.} \]
of a pendulum suspended from a hinge, which is intrinsically stable.

Figure 1 schematically depicts the working principle of the VPP-controller. When no torso torque is applied, the GRF is directed at the hip joint as the GRF is solely caused by the spring force, this is depicted in the figure 1 (left). In figure 1 (center and right) the VPP is not in line with the leg, therefore a torso torque needs to be applied about the hip. The torso torque produces an additional GRF at the foot, denoted by $F_T$. The GRF caused by the spring force ($F_s$) and the additional GRF caused by the torso torque ($F_T$) combine into the resultant force ($F$), directed at the VPP.

Maus et al. observed, in gaits of both humans and animals, that the GRFs during running also converge to a single point on the torso, this is shown in figure 2. Moreover, Maus et al. showed that VPP-control can stabilise the upright gait of a simple running model, in which the leg is modelled as a massless spring. A more recent study performed by Andersson continued on this by successfully implementing the VPP-controller on a running model with non-zero leg mass [16]. These two studies show that VPP control can be used to stabilise the torso, however they do not report on the control performance.

The objective of this paper is to determine the performance of the VPP-controller for a humanlike model. Therefore, we implemented the VPP control onto a realistic humanlike model, which includes a knee joint and a humanlike mass distribution. The performance of VPP control is compared against the more commonly used PD-control. The controller performance is expressed in terms of disturbance rejection, as this is a good indicator for the performance of the controller in real world situations.

The remainder of this paper is organised as follows: Section II provides detailed information on the implementation of the running models, the implementation of the controllers, and the used testing method. For both controllers, the maximum allowable disturbance and disturbance response are presented in Section III. The discussion can be found in Section IV, containing an interpretation of the results, a study on the influence of the model parameters and the limitations of this study. Finally, conclusions are drawn in Section V.

II. METHOD

This section describes the method that is used to quantitatively compare the PD-controller and the VPP-controller. The simple and the realistic running model used for the comparison are both described in Section II-A. The implementation of the controllers on the running models is described in Section II-B. The disturbance rejection of the torso pitch controllers in conjunction with the running models is numerically determined in simulation, this process is described in Section II-C.

A. Models

The performance of both controllers is tested using two running models: the simple running model and the realistic running model. The simple running model is based on the widely adopted Asymmetric Spring Loaded Inverted Pendulum (A-SLIP) model [17], [18], which is a simplified model of the dynamics of bipedal motion. Despite the simplicity of the simple running model it is believed that this model still features the same dynamic behaviour as observed in bipedal running [14]. In this study, the simple running model is used for computationally demanding simulations like gait optimisation problems.

The realistic running model is used for acquiring detailed and realistic results and for verifying the results of the simple running model. Both running models feature either the same or equivalent model parameters, this facilitates an inter model comparison of the results. The model parameters are based on normalised humanlike values. Table I and table II show the parameters that are used for the simple running model and the realistic running model respectively.

The Equations of Motion (EoM) of both models describe the movement of the two models in space. The EoM are implemented in Mathworks Matlab. The built-in ODE45 integrator function is used to numerically solve the EoM over each time step. The next subsections describe the simple running model and the realistic running model in detail.

1) Simple running model: Figure 3 depicts the simple running model. Except for the augmented torso it is equal to the commonly used SLIP model [17], [18]. The addition of a torso introduces torso pitch dynamics to the running model [19], making it useful for studying torso pitch controllers. The torso is modelled as a rigid mass $m$ with rotational inertia $I$ about its Center of Mass (CoM). The CoM of the torso of the simple running model has an offset $r$ from the hip joint. The leg is connected to the hip joint and is modelled as a massless spring with linear stiffness $k$ and rest length $L_0$.

The angle between the ground and the leg at the moment the foot touches the ground is called the angle of attack $\alpha_0$. The location of the model in space is described by the horizontal position $x$, the vertical position $y$ and the torso orientation $\phi$. These variables, in combination with their corresponding velocities $\dot{x}$, $\dot{y}$ and $\dot{\phi}$ are used to describe the state of the robot during running. The state of the model at touchdown is reduced to four state parameters only, as the vertical position $y$ can be
derived from the remaining states and the angle of attack. The horizontal position \( x \) is omitted because it is irrelevant for the motion of the model. The initial state of the simple running model is described as follows:

\[
\nu_0 = \left[ \varphi_0 \ x_0 \ y_0 \ \dot{\varphi}_0 \right]^T \tag{1}
\]

As depicted in figure 3, a single step of the simple running model consists of two phases, a flight phase and a stance phase. Touchdown (A) marks the beginning of the stance phase. The instant the foot loses contact with the ground is called lift-off (B), this moment marks the end of the stance phase and the beginning of the flight phase. During flight phase the foot is not in contact with the ground. The highest point during the flight phase is called apex (C). The instant the foot touches the ground is again called touchdown (A).

The simple running model is a conservative system, because no system energy is lost by impact or friction. Therefore, no actuation is needed to maintain the system at a constant energy level. During the stance phase, the torso pitch control can apply a torque about its hip joint. Exerting a torque about the hip causes the torso to rotate with respect to the leg. A torso torque can only be applied during the stance phase, because the lack of a GRF during flight would cause an infinite acceleration of the massless leg. The torso torque is limited by \( T_{\text{max}} \) to reflect the limitation of the human body. Furthermore, the torso torque is dynamically limited to prevent a negative GRF. A negative GRF causes the foot to lose its contact with the ground before the spring is elongated to its rest length \( L_0 \). This restriction of the torso torque for a stance leg length \( L_{st} \) is implemented through the following constraints:

\[
T_{\text{torso}} \leq T_{\text{limit}}, \text{ for } \alpha < 90^\circ \\
T_{\text{torso}} \geq T_{\text{limit}}, \text{ for } \alpha > 90^\circ
\]

with:

\[
T_{\text{limit}} = \tan(\alpha)F_sL_{st}
\tag{2}
\]

The simple running model assumes no slip of the foot during stance phase.

2) **Realistic model**: The realistic running model is created to simulate the dynamics of a running human. In this study, we focus on the movement of a runner in the sagittal plane, therefore the human body is modelled with a 2D model. The realistic model is built up out of two legs and a torso. Each leg consists of an upper and a lower leg, interconnected by a knee joint with a hyper extension stop. The knee joint contains a torsional spring with a stiffness profile chosen exactly such that the resulting stiffness \( k \) between the hip joint and the foot is linear. The torso is modelled as a distributed mass with an offset \( c_b \) between the CoM and the hip joint.

Figure 4 depicts the model parameters (left) and the state parameters (right) of the realistic running model. The state parameters together with their derivative counterparts define the exact state of the robot as it moves through space. At touchdown the state of the model is constrained by the constant initial forward speed \( \dot{x}_{h,0} \), the angle of attack \( \alpha_0 \) and the knee end-stop angle \( \varphi_{l,st,0} \), therefore the exact state of the robot at touchdown can be expressed by:

\[
\nu_0 = \left[ \varphi_{u,sw} \ \varphi_{l,sw} \ \varphi_{\text{torso}} \ \dot{y}_{\text{hip}} \ \dot{\varphi}_{u,sw} \ \dot{\varphi}_{l,sw} \ \dot{\varphi}_{\text{torso}} \right]^T
\tag{3}
\]

The model is actuated by torques about the hip and knee joints. Both hip torques act on the torso, therefore the resultant

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>( m )</td>
<td>71.4 kg</td>
</tr>
<tr>
<td>Torque limit</td>
<td>( T_{\text{max}} )</td>
<td>200 Nm</td>
</tr>
<tr>
<td>Spring rest length</td>
<td>( L_0 )</td>
<td>1 m</td>
</tr>
<tr>
<td>Spring stiffness</td>
<td>( k )</td>
<td>20000 N/m</td>
</tr>
<tr>
<td>Angle of attack</td>
<td>( \alpha_0 )</td>
<td>75(^\circ)</td>
</tr>
<tr>
<td>Rotational inertia</td>
<td>( I )</td>
<td>6.67 Nm/s(^2)</td>
</tr>
<tr>
<td>Initial speed</td>
<td>( x_0 )</td>
<td>3 m/s</td>
</tr>
<tr>
<td>Gravitational acceleration</td>
<td>( g )</td>
<td>9.81 m/s(^2)</td>
</tr>
</tbody>
</table>
torso torque $T_{\text{torso}}$ is defined as follows:

$$T_{\text{torso}} = -T_{\text{hip,sw}} - T_{\text{hip,st}}$$  \hspace{1cm} (4)

where:

- $T_{\text{hip,sw}}$ is the hip joint torque of the swing leg and
- $T_{\text{hip,st}}$ is the hip joint torque of the stance leg.

The actuation torques are limited by a fixed value $T_{\text{max}}$ to reflect the limitations of the human body. Additionally, the stance hip torque is dynamically limited by $T_{\text{lift},st}$ to prevent foot lift-off before the knee end-stop has been reached. Summarising, the actuation torque is limited according to:

$$|T_{\text{action}}| \leq \min(T_{\text{max}}, |T_{\text{lift},st}|)$$  \hspace{1cm} (5)

The realistic running model assumes no slip of the foot during stance phase.

The running cycle of the realistic running model is built up out of the following successive phases:

- **Stance phase:** The running cycle starts at the beginning of the stance phase. The initial state $\nu_0$ describes the state parameters as described in (3). During stance phase, one foot is in constant contact with the ground, while the stance leg flexes about the knee joint. As the stance leg flexes, potential energy is stored in the torsional spring in the knee joint, subsequently this elastic energy is released as the leg extends. The end of the stance phase is reached when the leg is fully extended. This event triggers the next phase, the knee impact.

- **Knee impact phase:** The knee end-stop prevents over-extension of the stance leg. The impact of the lower leg against the knee end-stop is modelled as an instantaneous impact. The model lifts off after knee impact, marking the transition to the flight phase. After knee impact the stance and swing leg swap.

- **Flight phase:** During the flight phase, there is no contact with the ground. The upcoming stance leg is controlled along a quintic spline trajectory towards the preset angle of attack. The instant the foot hits the ground is called touchdown, this triggers the next phase: the foot impact phase.

- **Foot impact phase:** This phase simulates the instantaneous change in velocity caused by the impact between the foot and the ground. The foot impact marks the end of one running cycle.

Both the knee impact and foot impact reduce the system energy and therefore make a stable gait impossible. To maintain the system at a constant energy level, a push-off with a constant amount of kinetic energy is exerted during extension of the leg. This push-off is implemented as knee torque that is applied during a fixed range of knee angles. The knee torque is exerted in the same direction as $\phi_{l,st}$, when the stance leg extends. The initial forward speed of the hip joint is set to $3 \text{ m/s}$. Quintic splines are used to describe smooth trajectories between the current swing leg position and the desired angle of attack. Low level PD-control governs tracking of these trajectories by exerting torques about the applicable joints. Fixing the angle of attack in conjunction with a constant push-off and a fixed initial forward speed constrains the model towards a humanlike running gait.

![Realistic running model](image)

Figure 4: The realistic running model. **Left:** model parameters of the realistic running model. The realistic model features two legs and a torso. Each leg consists of an upper and a lower leg, interconnected by a knee joint. The legs and torso contain both rotational inertia and mass. Realistic model parameters are shown in Table II. **Left:** state parameters of the realistic running model. $\phi_{u,sw}$ and $\phi_{u,st}$ denote the upper swing leg angle and the upper stance leg angle respectively. $\phi_{l,sw}$ and $\phi_{l,st}$ denote the lower swing leg angle and the lower stance leg angle respectively. The torso orientation is defined by $\phi_{\text{torso}}$. Adapted from [20], [21].

### Table II: Realistic running model parameters.

<table>
<thead>
<tr>
<th></th>
<th>upper body</th>
<th>upper leg</th>
<th>lower leg</th>
</tr>
</thead>
<tbody>
<tr>
<td>mass $m$ [kg]</td>
<td>48.2</td>
<td>7.3</td>
<td>4.3</td>
</tr>
<tr>
<td>mom. of inertia $I$ [kgm$^2$]</td>
<td>2.1</td>
<td>0.12</td>
<td>0.19</td>
</tr>
<tr>
<td>length $l$ [m]</td>
<td>0.56</td>
<td>0.48</td>
<td>0.56</td>
</tr>
<tr>
<td>vert. dist. CoM $c$ [m]</td>
<td>0.35</td>
<td>0.19</td>
<td>0.36</td>
</tr>
<tr>
<td>hor. offset CoM $w$ [m]</td>
<td>0.00</td>
<td>-0.01</td>
<td>-0.02</td>
</tr>
</tbody>
</table>

### B. Controller Implementation

Both VPP control and PD-control attempt to stabilise the torso by exerting a torso torque about the hip joint. This section elaborates on the implementation of these controllers onto the two simulation models. During flight phase no GRF is present, therefore both controllers can exert a torso torque only during the stance phase.

1) **VPP-controller implementation:** VPP control redirects the GRF at the VPP by applying a torque about the hip. The implementation of VPP control into the simple running model and the realistic running model differs, as the simple running model features a massless leg, whereas the legs of the realistic model do contain mass.

**Simple running model:** For the simple running model without torso torque, the GRF is caused by the spring force only. Without torso torque, the magnitude of the GRF is equal to the spring force and by default directed at the hip joint. A torso torque creates an additional GRF perpendicular to the spring force, causing the resultant GRF to change direction and magnitude. The torso torque that is needed to redirect the GRF at the VPP is determined as follows:

$$T_{\text{torso}} = k(L_0 - L_{st})L_{st} \frac{(r + Q)\sin\phi}{L_{st} - (r + Q)\cos\phi}$$ \hspace{1cm} (6)

In which $L_{st}$ is the leg length. The CoM offset $r$, the spring constant $k$ and the spring rest length $L_0$ are model parameters.
The model parameters of the simple running model can be found in table I.

**Realistic running model:** The implementation of the VPP controller on the realistic model is not as straightforward as on the simple running model, because the direction and magnitude of the GRF is influenced by the additional leg inertia and knee joint. Three methods have been identified to implement the VPP-controller onto the realistic running model:

Method 1: Numerically solve the EoM

This implementation calculates the exact torso torque needed to direct the GRF at the VPP. First, the EoMs are used to evaluate the initial direction and magnitude of the GRF when no torso torque is applied. Subsequently, the magnitude and direction of the desired GRF can be calculated based on the current location of the VPP and the initial GRF. Finally, the EoMs are reversed solved to calculate the hip actuation torque needed to produce the desired GRF. Although this implementation results in the exact torso torque output that is needed to direct the GRF at the VPP, it does have two disadvantages. Reversely solving the EoMs is a computationally demanding calculation, which cannot be performed in real time. Furthermore, for application on a real world robot this method is bound to rely on a simulation model for determining the unactuated GRFs, as for a real robot it is impossible to measure the unactuated GRF while simultaneously controlling the torso torque.

Method 2: Feedback loop

The second implementation of the VPP-controller relies on a feedback loop with a P-control scheme to minimise the error between the direction of the GRF without torso torque and the desired direction of the GRF. The torque output is calculated by multiplying the error with gain \(K_p\):

\[
T_{\text{torso}} = K_p \gamma_{\text{GRF}} - \gamma_{\text{VPP}}
\]

where:
- \(K_p\) is the error gain,
- \(\gamma_{\text{GRF}}\) is the GRF angle without torso torque and
- \(\gamma_{\text{VPP}}\) is the VPP angle.

An advantage of this implementation in simulation is the low computational power needed to calculate the torso torque as all components of (7) are readily available in the simulation environment. However, the error gain \(K_p\) influences the calculated value of \(T_{\text{torso}}\) and therefore needs to be included in the optimisation of the controller parameters. Furthermore, determining the GRF-angle on a real robot without torso torque cannot be done during running. Therefore, the implementation of this method on a real robot might involve using an additional computational model for real-time calculation of the GRF.

Method 3: Simple model method

A third implementation method is to neglect the effects of leg mass, leg inertia and a knee joint on the magnitude and direction of the GRF. This simplification makes it possible to use the same implementation on the realistic model as is used on the simple running model. The disadvantage of this method is that the calculated torso torque will not exactly direct the GRF at the VPP. Section IV will discuss the consequences of this simplification. The advantage of this implementation is that it is not computationally intensive. Due to the limitations of implementation method 1, only implementation method 2 and implementation method 3 are implemented on the simulation models.

2) PD-controller implementation: The PD-controller uses the current torso pitch angle \(\phi\) and the current torso pitch angular velocity \(\dot{\phi}\) to calculate the output torso torque \(T_{\text{torso}}\):

\[
T_{\text{torso}} = -K_p^{\text{PD}} (\phi - \phi_{\text{des}}) - K_d^{\text{PD}} \dot{\phi}
\]

where:
- \(K_p^{\text{PD}}\) is the positional gain,
- \(K_d^{\text{PD}}\) is the velocity gain,
- \(\phi\) is the current torso pitch angle,
- \(\dot{\phi}\) is the current torso angular velocity and
- \(\phi_{\text{des}}\) is the desired torso pitch angle.

Deviations from the desired value for both the torso pitch angle and the torso angular velocity are multiplied by the positional and the velocity gain respectively. The sum of these components is equal to the torso torque output.

**C. Disturbance rejection**

The disturbance rejection is quantified by two performance indicators: maximum allowable disturbance and step response. This section describes the method that is used to determine the maximum allowable disturbance rejection of the VPP-controller and the PD-controller. A schematic representation of this method is depicted in figure 5.

**Determine Limit cycle:** The initial conditions define the set of state parameters for which the running model is in limit cycle, which means the runner is in a nominally periodic, stable sequence of hops. The initial state of a consecutive hop is determined by the end state of the previous hop. The mapping between the initial states of consecutive hops is called a Poincaré map, this can be denoted as:

\[
\nu_{n+1} = S(\nu_n)
\]

In which \(\nu_n\) is the initial state as described in (1) for the simple running model and (3) the initial state for the realistic model. The function \(S\) maps the state \(\nu_n\) at touchdown to the initial state at next touchdown \(\nu_{n+1}\). Function \(S\) is called the stride function and it can be obtained by numerically integrating
the EoM over one hop. The model is in limit cycle when the initial states remain unchanged for consecutive hops, this means the model is going through a periodic motion [21]–[24], in accordance to:

\[ \nu_{n+1} = \nu_n \]  

(10)

For a set of simulation parameters and model parameters a limit cycle is found by minimising the difference between the initial state \( \nu_0 \) and the state of the running model after one step \( \nu_1 \). This optimisation is automated in fminsearch, a built-in optimisation function from the Matlab toolbox. A running cycle is influenced by changing model parameters or torque actuation, therefore a limit cycle is unique to each specific combination of model parameters and controller actuation. For PD-control, the desired torso angle is optimised to find a solution for (9). For the VPP-controller on the realistic model, the VPP-distance is optimised when determining the limit cycle. Maus et al. show that the most stable limit cycles could be found for a VPP-offset ranging from 0.05m to 0.5m [14]. The VPP-distance is therefore lower and upper bound in optimisation by 0.05m and 0.5m respectively.

**Run 25 steps:** The simulation models are used to test if the model is able to run upright for a combination of initial conditions, controller parameters and model parameters. It is assumed that the model is able to continue running for an infinite number of steps when it is able to complete 25 consecutive steps, therefore a threshold of 25 steps is used to define successful running.

Because the EoM cannot be solved analytically, the motion of the model is calculated by numerically solving the EoM for numerous of small consecutive time steps. The simulation is stopped when the model either completes 25 consecutive steps or when the model falls down. The fall-down event is triggered when the height of the hip joint is equal to zero or when a negative GRF is detected during stance phase.

A higher level control is used to stabilise the model over multiple steps by adjusting the angle of attack \( \alpha_0 \) and thereby controlling the forward speed [25]. The angle of attack at step \( n \) is determined by:

\[ \alpha_n = \alpha_0 + K_{\alpha}(\dot{x}_0 - \dot{x}_n) \]  

(11)

The nominal angle of attack and the nominal forward speed are denoted by \( \alpha_0 \) and \( \dot{x}_0 \) respectively. For the simple running model the nominal angle of attack is 75°. For the realistic running model, the nominal angle of attack differs, as it is included in the limit cycle optimisation. For both models, the value of the angle of attack gain \( K_{\alpha} \) is determined experimentally.

**Increase disturbance:** Disturbances are introduced to the model as a single error on the initial state of the limit cycle. The magnitude of the error is increased until the running model falls down. The initial state is subject to three types of disturbances:

**Horizontal velocity:** The disturbance in the horizontal velocity is modelled as a deviation from the initial condition \( \dot{x}_0 \) of the nominal limit cycle. This disturbance equals an instant speed increase or decrease of the CoM. The horizontal velocity disturbance is increased in steps of 0.05\( \text{m/s} \) and upper bound at 2.5 \( \text{m/s} \).

**Floor height:** The floor height disturbance is modelled as a deviation from the initial condition \( y_0 \) of the nominal limit cycle. The magnitude of the floor height disturbance relates to \( y_0 \) as follows:

\[ y_{\text{floor}} = \frac{\dot{y}_0^2}{2g} \]  

(12)

\( y_{\text{floor}} \) is analogous to running up a step. After the successful completion of 25 hops the step down disturbance is increased in steps of 0.001 \( m \) and is upper bound at 0.5 \( m \).

**Torso angular rate:** The disturbance on the angular rate of the torso is modelled as a deviation from the initial condition \( \dot{\phi}_0 \) of the nominal limit cycle. After the successful completion of 25 hops the torso angular rate is increased in steps of 3.6°/\( s \). The disturbance is upper bound at 180°/\( s \) to prevent unrealistically high disturbances.

**Adjust controller parameters:** As described in Section II-B, both controllers feature parameters that influence the torque output and thereby change the controller performance. The performance of both controllers is determined by the best performance of the full range of controller parameters. Therefore, the maximum allowable disturbance is determined over a range of controller parameters.

For the PD-controller a positional gain and a velocity gain is used to determine the torque output. To maximise the performance of this controller, while avoiding a time-consuming optimisation, the ratio between the positional and velocity gain is fixed. Based on the heuristic Ziegler-Nichols method, the following ratio between proportional and derivative gain leads to near optimal controller performance for this control system:

\[ K_{P\alpha} = \frac{K_{P\alpha}}{10} \]  

(13)

A fixed gain ratio reduces the search space of optimal controller settings to only one parameter. In Section IV the validity of this simplification is discussed.

**III. RESULTS**

In this section the results of the simulation study are presented. In Section III-A the maximum allowable disturbance of both controllers is presented. In the comparison of both controllers, the maximum allowable disturbance is used as the performance measure, as the maximum allowable disturbance resembles realistic situations that a real robot encounters while running. In Section III-B the disturbance response of both controllers is presented. The disturbance response provides information on the recovery process of the model after a disturbance.

**A. Maximum disturbance**

Figure 6 shows the maximum allowable disturbance of the PD-controller and VPP-controller for a range of controller parameters. The maximum positive and negative allowable disturbance is determined for three different types of disturbances: forward velocity disturbance (top row), floor height disturbance (center row) and angular torso velocity disturbance.
Figure 6: Disturbances for which the PD-controller and the VPP-controller are able to complete 25 steps. The left sub-plots show the maximum allowable disturbances of implementation method 2 of the VPP-controller versus the VPP error gain parameter. The right sub-plots show the maximum allowable disturbances of the PD-controller versus the P-gain. The red dots mark the controller parameters for which the highest maximum allowable disturbance is attained on the realistic running model. The results for the VPP-controller on the realistic running model were acquired using implementation method 2 from Section II-B1. The initial, undisturbed forward velocity is 3 m/s, with a 0.08 [°] angle of attack gain.
(bottom row). The left-hand plots show the results of the VPP-controller, whereas the right-hand plots show the results of the PD-controller. The green surface represents the maximum allowable disturbance that was calculated using the realistic running model, whereas the dotted blue lines represent the maximum allowable disturbances of the simple running model. A maximum allowable disturbance of zero means that the model is not able to run 25 steps for that particular controller parameter, in this case the maximum allowable disturbance is zero for all 3 types of disturbances.

As discussed in II-B1, implementation method 2 is used to implement the VPP-controller on the realistic running model. According to (7), this implementation method features a feedback gain, therefore the maximum allowable disturbance of the VPP-controller of the realistic running model is shown for a range of P-gains. The implementation of the VPP-controller on the simple running model does not feature a controller parameter, therefore the maximum allowable disturbance of the VPP-controller on the simple running model is shown as a horizontal dotted blue line with a constant maximum allowable disturbance.

The red dots in the figure mark the controller parameters for which the highest maximum allowable disturbance is attained on the realistic running model. Comparing the performance of the PD-controller to the performance of VPP-controller is done by comparing the maximum allowable disturbance attained at the red dots. The red dots show that the PD-controller is able to handle disturbances on both forward velocity and floor height disturbance of slightly greater magnitude. For the disturbances on the angular velocity of the torso this is different, as for this type of disturbance the PD-controller is able to withstand positive disturbances of $+43.2\,^\circ/s$ and negative disturbances of $-122\,^\circ/s$, whereas the VPP-controller is able to handle a positive and negative disturbances of $14.4\,^\circ/s$ and $-14.4\,^\circ/s$ respectively. From the results of the realistic model can be concluded that the PD-controller matches or outperforms the VPP-controller in terms of maximum allowable disturbance over all types of disturbances.

From the maximum allowable disturbance of the PD-controller on both the simple and realistic running model (right-hand plots) it is observed that the PD-controller is able to handle larger disturbances on the simple model than on the realistic model. Furthermore, the optimal P-gain for the realistic model is within the 500 to 1500 range, depending on the type of disturbance. For the simple running model, a P-gain of 3500 leads to optimal performance, as the highest maximum allowable disturbance is attained for this P-gain. For the simple running model, a P-gain exceeding 3500 does not lead to a higher maximum allowable disturbance as for these P-gains the effective torso torque output is capped by the torque limiter, as described by (5).

B. Disturbance response

According to (10), the model is running in limit cycle if the state $v_n$ of the model at touchdown is the same for each consecutive step $n$. The space spanned by all states, for which the model is able to run, is called the Basin of Attraction (BoA). When the model is disturbed out of its limit cycle, but still within the BoA, it will continue running. After such a disturbance, the model will return to a limit cycle within the BoA. The behaviour of the model after such a disturbance is called the disturbance response, it shows how fast the model returns to a limit cycle. In accordance to [26], the step time is used to indicate whether the running model has already returned to its nominal limit cycle.

Figure 7 shows a typical disturbance response of the PD-controller and the VPP-controller on the realistic running model. For both controllers on the realistic running model the step time is shown over the course of 35 steps. At step 1 the realistic running model is initialised in its limit cycle, at step 5 the model is disturbed from its limit cycle. This disturbance is introduced to the model as a single positive error on the nominal forward velocity at the moment of touchdown. The dotted horizontal lines represent a deviation of 2% from the original step time.

![Figure 7: A typical disturbance response of the PD-controller and the VPP-controller on the realistic running model. The normalised step time is shown for 35 consecutive steps. The first 5 steps the model is running in limit cycle, at step 5 a single disturbance of $+0.15\,m/s$ is introduced to a nominal forward velocity of $3\,m/s$. The PD-gain is 900 [-], the VPP-gain is 2250 [-].](image)

Figure 7 shows that the settling time of the PD-controller is smaller than that of the VPP-controller, as the PD-controller needs 5 consecutive to return within a 2% margin of its initial step time, whereas the VPP-controller needs 16 steps. Furthermore, the VPP-controller shows a maximum deviation of the step time of 9%, whereas the maximum deviation of the step time of the PD-controller is 4%. From step 7 on, the VPP-controller shows an oscillation of the step time, whereas only one oscillation of the step time is observed for the PD-controller.

IV. Discussion

In the previous section two torso controllers have been compared in terms of maximum allowable disturbance and
disturbance response. In Section IV-A an interpretation of the results is given. The results were acquired for one set of simulation parameters, Section IV-B investigates the influence of simulation parameters variations. Finally, the limitations of this simulation study are discussed in Section IV-C.

A. Interpretation

The maximum allowable disturbances of the VPP-controller on the simple running model are matched or exceeded by the PD-controller. Furthermore, the results that were acquired on the realistic model validate the results of the simple running model, as the VPP-controller is also outperformed by the PD-controller on the realistic model. The difference between the performance of the PD-controller and the VPP-controller can be explained by two reasons.

Firstly, the performance of the VPP implementation methods is investigated. The objective of the VPP-controller is to direct the GRF at the VPP by applying a torso torque. Implementation method 2 features a gain to minimise error between the actual GRF-direction and the desired GRF-direction. Implementation method 3 neglects the effects of leg mass, leg inertia and knee joints on the magnitude and direction of the GRF. The performance of both implementation methods is shown in figure 8. The path of the VPP during the stance phase of the realistic model is depicted by the dashdot line and the dotted line respectively. The figure shows an offset between the desired GRF-angle and the actual GRF-angle of both methods. A contribution of the torque limit to the GRF-angle offset is expected, as in some model positions the torque required to direct the GRF at the VPP can exceed the maximum torso torque output.

Both methods do not succeed to accurately direct the GRF-angle at the VPP, and thereby depreciate the performance of the VPP-controller. Furthermore, implementation method 3 did not lead to stabilised running as no limit cycle was found. Implementation method 1 was too computationally expensive and was therefore not used in the comparison.

Secondly, from the disturbance response of both controllers (figure 7), it is observed that the VPP-controller produces an oscillation in step time during the recovery from a disturbance, whereas the PD-controller does not show this behaviour. The cause for this behaviour is found when investigating both control schemes. The PD-controller features a damping term, whereas the VPP controller determines the control output on positional information only. From this is concluded that the lack of damping of the VPP-controller is the cause for reduced performance. Future research is needed to investigate if the addition of a damping term to the VPP control scheme improves performance.

For the PD-controller we have seen that the optimal P-gain for the simple running model is much higher than the optimal P-gain for the realistic running model. The explanation for this behaviour is found by investigating the weight distribution of both running models. For the simple running model the total mass is concentrated solely on the torso whereas the legs of the realistic running model do feature mass. Therefore, a larger torso torque is needed to keep the torso of the simple running model upright during running. For PD control, higher torso torques are attained at higher controller P-gains.

The disturbance response of both controllers (fig. 7) shows peaks in both lines, representing the maximum deviations from the initial step time of the VPP-controller and the PD-controller. For these particular controller parameters, the PD-controller is able to withstand a maximum disturbance of 0.25 m/s, whereas the VPP-controller is able to withstand a maximum disturbance of 0.15 m/s on the forward velocity. Therefore the different peak size is expected, as the VPP-controller is disturbed with its maximum allowable disturbance, whereas the PD-controller is able to handle 40% greater disturbances.

B. Parameter variation

The simulation was run with a single set of model and simulation parameters. This raises the question if the conclusions are generic over a range of parameters, or if they are only valid for a single set. Therefore, the parameters most likely to influence the controller performance are investigated.

Torso CoM-offset: Robots are often designed to have their torso CoM coinciding with the hip joint, allowing for simple torso controllers. Recently, van Oijen et al have shown that a larger step-down disturbance can be handled by models with their CoM above the hip [27]. To investigate the influence of the CoM-offset on the maximum allowable disturbance, the simulation is rerun for a realistic range of CoM-offsets, ranging from 0 m to 1 m.

In figure 9 the maximum allowable disturbance is shown over a range of CoM-offsets. For each CoM-offset, the maximum allowable disturbance is shown for the best performing
Figure 9: Maximum allowable disturbance for a torso CoM-offset ranging from 0 m to 1 m, sampled with 0.175 m increments. The maximum allowable disturbance is shown for the best performing controller parameter only. The left plots show the allowable disturbances of the VPP-controller, the right plots show the allowable disturbances of the PD-controller. Results were acquired on the realistic running model for an initial, unperturbed forward velocity of 3 m/s, with an 0.08 [–] angle of attack gain. The tested forward velocity disturbance is limited to 2.5 m/s and the angular velocity disturbance is limited to 180°/s.
The vertical red dotted lines depict the maximum allowable disturbance for a humanlike CoM-offset of 0.35 m, as was used in figure 6. The left-hand plots show the maximum allowable disturbance over a range of CoM-offsets of the VPP-controller, the right-hand plots show the maximum allowable disturbance of the PD-controller. Comparing the maximum allowable disturbance of the PD-controller and the VPP-controller, it is observed that the PD-controller outperforms the VPP-controller over the full range of tested CoM-offsets, thereby supporting previous conclusions. Interestingly, these results do not support the findings of van Oijen et al, as the results show that the largest floor disturbances were handled by the realistic running model with a zero CoM-offset. The exact reason for this difference is unknown, but it can possibly be explained by the difference between the used running models. The realistic running model used in this study does feature leg mass, whereas van Oijen et al used a simulation model without leg mass.

**PD-control gain:** To limit the number of PD control parameters, the velocity gain was fixed to 1/10th of the positional gain, according to (13). To verify the validity of this simplification, the simple running model is used to determine the maximum allowable angular disturbance over a range of positional gains and velocity gains. Figure 10 depicts the gain.

The figure shows that for an arbitrary value of the proportional gain, a 10 times smaller derivative gain does maximise the maximum allowable disturbance. This observation justifies the use of a constant ratio between the proportional and the derivative gain, as was proposed in (13).

**C. Limitations**

As often the case for simulation studies, some remarks regarding the limitations of this study are in place. The focus of the study is on the performance of the torso controllers, therefore a simple, higher level angle of attack controller was used to control the forward velocity. Due to the complex running dynamics, it is unknown how the angle of attack controller and the torso controller interact during running. Therefore, more research on the effects of the angle of attack control on the torso control is needed.

Furthermore, the realistic running model was used to acquire accurate data of a running model. In the creation of the realistic running model, simplifications have been made. Saturation of the actuators is implemented as a constant limit on the torque output and no slip of the foot on the ground is assumed. Even though the realistic running model does seem to exhibit a humanlike running gait, future research is needed to verify the results of this study. A real robot can be used to verify the validity of the running model.

In the comparison of this study, performance was quantified in terms of maximum allowable disturbance. Future research may include other performance indicators like energy consumption, controller robustness and stability.

**V. Conclusions**

In this paper two torso pitch controllers were compared in terms of maximum allowable disturbance and disturbance response. From this study it can be concluded that:

- PD-control handles larger disturbances than VPP-control.
- When disturbed, PD-control returns the running model to its limit cycle faster than VPP-control.
- Small CoM-offsets lead to higher maximum allowable disturbances.

**References**


