

Real-Time Joint Coupling of the Spine for Inverse Kinematics

Daniel Raunhardt and Ronan Boulic

VRLab

Faculte informatique et communications

Ecole Polytechnique Federale de Lausanne (EPFL)

EPFL IC ISIM VRLAB, Station 14, 1015 Lausanne, Switzerland

email: daniel.raunhardt, ronan.boulic@epfl.ch

www: <http://vrlab.epfl.ch>

Abstract

In this paper we propose a simple model for the coupling behavior of the human spine for an inverse kinematics framework. Our spine model exhibits anatomically correct motions of the vertebrae of virtual mannequins by coupling standard swing and revolute joint models. The adjustment of the joints is made with several simple (in)equality constraints, resulting in a reduction of the solution space dimensionality for the inverse kinematics solver. By reducing the solution space dimensionality to feasible spine shapes, we prevent the inverse kinematics algorithm from providing infeasible postures for the spine. In this paper, we exploit how to apply these simple constraints to the human spine by a strict decoupling of the swing and torsion motion of the vertebrae. We demonstrate the validity of our approach on various experiments.

Keywords: Inverse Kinematics, joint coupling, virtual human modeling, articulated structure

Digital Peer Publishing Licence

Any party may pass on this Work by electronic means and make it available for download under the terms and conditions of the current version of the Digital Peer Publishing Licence (DPPL). The text of the licence may be accessed and retrieved via Internet at <http://www.dipp.nrw.de/>.

First presented at the International Conference on Computer Graphics Theory and Application (GRAPP) 2007, extended and revised for JVRB

1 Introduction

Realistic animations of human characters play an important role in virtual environments. The common way to let a user interact with the virtual environment is the use of a three dimensional virtual mannequin that is controlled by the user [MTT04][PZB90]. The more realism of the virtual mannequin and the virtual environment, the more the user feels present in the virtual environment [SUS95]. Often researchers have only improved the representation and animation of the surface such as exterior skin or muscle deformations while the underlying body model has been kept unchanged [WG02]. However, increasing the surface details of the virtual mannequins may lead viewers to be more sensitive to unrealistic joint motions in animations such as in the shoulder and spine regions of the human body [HOT98].

Joint models are important for correct motion analysis of tasks, including the estimation of muscle lengths and moment arms [DL95]. Introducing accurate biomechanical joint models to the traditional hierarchy of joint transformations can lead to an improved realism in the human character animation. The anatomical correct axes of rotations and joint centers are taken from literature and refined to produce improved fidelity of the motions.

In this paper we introduce a human spine model based on a set of coupled vertebral joints. These vertebral joints are coupled through simple linear (in)equality constraint reflecting their anatomic mobility distribution [Kap87][Kap82]. We will show how these constraints transparently can be integrated into an existing inverse kinematics (IK) solver. A few parameters will be enough to control the complex articulation of

the human spine as each equality constraint reduces the solution space for IK solver at least by one dimension. We give an introduction to our IK solver that is based on Prioritized Inverse Kinematics [BB04]. We demonstrate the validity of our spine model with different experiments including performance measurements.

In the next section, we review previous work. Section 3 gives an overview of our PIK solver and presents how we can couple the joints by constraining the solution space. Section 4 details our spine model. Section 5 presents several applications and experimental results. Finally, we present future work and conclude this paper in Section 6.

2 Previous Work

The first use of articulated joint models for representing human joints can be found in the studies of kinematics of robotic manipulators. These systems used the Denavit-Hartenberg link parameter notation from robotics to represent virtual mannequins with articulated limbs [GM98]. The Denavit-Hartenberg notation is used for linking parameters by associating coordinate frames between adjacent segments. This is convenient but each parameter can only describe one degree of freedom (dof) between two adjacent segments. By combining sets of parameters also multiple dofs could be achieved.

Some regions of the human body exhibit a high coupling behavior of the joints like the human shoulder or spine. The shoulder of virtual humans is often modeled using three independent segments: the clavicle, the scapula and the arm using rotational joints. However, such a model is not sufficient for generating realistic motions. The reason is that such a joint model does not reflect the simultaneous motion of the joints [MT00]. Another strong coupling behavior can be observed in the human spine. The coupling phenomenon emerges from many factors, including the geometry of the vertebrae, the ligament that bind them, and the curvatures of the vertebral column. As an illustration of coupled vertebral motion, when a physician introduces a demand for rotation right to a patient in a neutral seated position, the response of the patients vertebrae involves coupled side-bending to the left (Figure 1).

This coupling behavior of the spine has been exploited by [MB91] to develop a kinematic model of the spine that exhibits flexion-extensions (forward-backward), lateral bending (left-right) and axial tor-

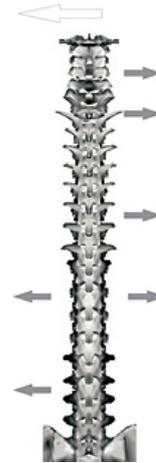


Figure 1: Regional Coupling Patterns: This diagram summarizes the coupling of lateral bending to the left (indicated by white arrow to the left). In the middle and the lower thoracic spine, the axial rotation is coupled with lateral bending that can be either to the left or to the right. In the lumbar spine, there is left lateral bending with left lateral bending while in the upper thoracic and cervical there is left lateral bending coupled with right lateral bending.

sion rotation. Another coupling approach for a normalized representation of the human skeleton of the spine has been presented by [KMA05]. The spine is modeled using a spline that could be divided into segments. The position of a vertebrae is given by discretizing the spline accordingly to the distances of the vertebrae. The Peabody system [MB91] collects joints into different groups of joints that have group angles to configure the joint groups' segments. [SNTH03] introduced a general joint component model called *joint maps* that allows modeling of joint expressions over several bone segments for biomechanical accurate joints. A joint map is a function that takes a set of inputs (e.g. set of joints) to produce an output for one or more joints. The input can be seen as the dofs and the outputs are the modified joint values (e.g. angle or translation variations). The outputs of one joint map can be combined with the inputs of another joint map to create increasingly sophisticated behaviors. The process that performs this mapping varies according to the type of the desired joint behavior (e.g. different coupling behavior of the shoulder and the spine). [HUF05] characterize the joint coupling behavior by implicit surfaces obtained from motion captured values. This representation allows to characterize intra- and inter-joint dependencies

but is not a very intuitive way to control the human motion of the joints.

Beside modeling the coupling behavior of the spine many approaches have been presented for biomechanical accurate joint models. To restrict the joint range angles spherical polygons [Kor85] and joint sinus cones [MT00][WG02] have been introduced. Although spherical polygons are more general than joint sinus cones, their handling is also more complex and for modeling the joint limits of human joints, the use of joint sinus cones is normally sufficient [Kor85]. [MNF02] incorporate joints that can translate and rotate together on a plane. The joint limits dynamically change with the dofs of the joints.

There is not one joint model that would be suitable for all these different characteristics. Often specialized joint models are used.

Our spine model is based on common joint models such as revolute (1 dof) or swing joint (2 dof) regularly used in IK. With simple linear (in)equality constraints we are able to group joints into sets and reducing simultaneously the solution space for the IK solver.

3 Prioritized Inverse Kinematics

3.1 Constraining the Solution

We provide here only a brief overview of the Prioritized Inverse Kinematics algorithm (PIK) [BB04]. The PIK algorithm relies on an efficient computation of projection operators, ensuring that a lower priority task cannot disturb any higher priority task. It can handle an arbitrary number of strict priority levels. In addition, our PIK solver can be augmented by linear (in)equality constraints for the purpose of controlling virtual mannequins or robots. In the specific case of the human body we are compliant with the Humanoid Animation Working Group standard(H-Anim) that includes all human joints [HA99].

In the normal case, the joints are independent of each other. Possible coupling due to the presence of tendons, muscles spanning over several joints or the above mentioned coupling of the spine or shoulder has to be defined as additional *hard* constraints which are also called *equality* constraints. Each equality constraint is defined by a simple linear equality equation and ensures that the corresponding parameters act accordingly to some given ratios defined in the equation. We also exploit inequality constraints for two purposes:

- Modeling joint ranges: to model anatomical correct joint we want to define joint limits that should never be passed.
- Offering a "relaxed" coupling behavior between joints. See Section 4.2

The equality constraint (Figure 2 left) and the inequality constraint (Figure 2 right) for a joint configuration q with n dofs are defined as follows:

$$c_i^T q = b_i \quad i \in 1..g \quad (1)$$

$$c_i^T q \leq b_i \quad i \in 1..g \quad (2)$$

where the c_i are n -dimensional vectors, b_i are scalars and g is the number of (in)equality constraints. The Equation (2) allows simple lower and upper bounds on joint variables for modeling joint limits as well as a relaxed coupling behavior. A set of linear inequality constraints defines a convex space of feasible joint configurations as seen in Figure 2 right. We have chosen linear (in)equality constraint by the higher complexity introduced by non-linear constraints. Generally, we model the joint coupling by equality constraints while we use for the joint limits inequality constraints.

The management of the (in)equality has somehow to be integrated within the process that computes the joint variation. Ignoring the constraints for the actual joint variation computation and only adjusting the resulting joint configuration to satisfy the constraints leads to non-optimal solutions [BB04]. The equality constraints can be ensured within a single iteration step during the initialization phase of the joint variation computation. To achieve this we modify the initial projection operator P_0 so that the resulting solution space lies within the constrained subspace. Secondly, the initial joint variation Δq^0 is set to the displacement required to meet the constraints, if this is not already the case. Often, the initial joint variation Δq^0 is zero as the constraints are met at the initialization phase. By modifying this initial joint variation Δq^0 , it helps also to avoid numerical drifts away from the constraints. The modified, initial projector operator P_0 remains constant as long as no equality constraint is added or removed. Thus, we can pre-compute P_0 and have only to modify it when an equality constraint is added or removed.

Compared to the PIK algorithm described in [BB04], these additional *hard* constraints can be seen as tasks of *infinite* priority (i.e. of higher importance than any

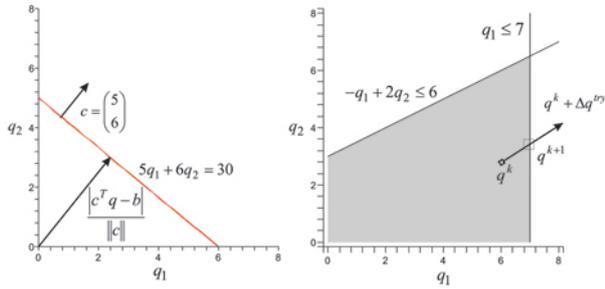


Figure 2: Left: equality constraint between two joints q_1 and q_2 . Right: the grey region indicates the possible area of solutions for the joint q_1 and q_2 . The computed solution $q^k + \Delta q^{try}$ violates the inequality constraint $q_1 \leq 7$ so the corrected solution is clamped to q^{k+1} .

other task) as they are shaping the initial solution space.

Algorithm 1 Projector Initialization.

- 1: $P_0 = I_n$
 - 2: **for all** $c_i^T q = b_i$ such that $i \in 1..g$ **do**
 - 3: **if** $\|P_0 c_i^T\|_2 = 0$ **then**
 - 4: $P_0 = P_0 - (P_0 c_i^T)(P_0 c_i^T)^T$
 - 5: **end if**
 - 6: **end for**
-

Algorithm 1 presents the procedure of the pre-computation of P_0 for the equality constraints. P_0 is initialized with the identity matrix I_n where n is the number of equality constraints. We are adding iteratively each equality constraint to P_0 only if the new constraint is not conflicting with the actual P_0 . An equality constraint is in conflict with P_0 when the new constraint is linear dependent which is the case whether the norm of the projection of the current constraint with respect to P_0 is not 0.

Figure 2 left gives a simple example for an equality constraint where q_1 and q_2 are two joints that are coupled by one equality constraint. This constraint ensures that the joint values q_1 and q_2 always lie on the line defined by $5q_1 + 6q_2 = 30$. Thus, the ratio between q_1 and q_2 remains constant. In the general case, the vector c defines a n -dimensional hyperplane on which the constrained joints lie.

The inequality constraints are checked *after* the new joint configuration q^{k+1} is computed, where the new joint state is defined as the sum of the current configuration and the computed solution

$q^k + \Delta q^{try}$. Figure 2 right illustrates a case where the new configuration $q^{k+1} = q^k + \Delta q^{try}$ would violate an inequality constraint. Therefore, dynamically a new equality constraint is added into the existing constraint set (only if this new constraint would not be conflicting with the other constraints). This new equality constraint ensures that the corresponding joint configuration q^{k+1} is *clamped* on the limit defined by the inequality constraint.

In our model, we check first the user-defined inequality constraints and afterwards the joint limits which are also modeled as inequality constraints. If two inequality constraints are conflicting, i.e. they cannot be satisfied at the same time, we consider only the constraint that has been first added to set of constraints. Thus, the user defined inequality constraints have a higher priority than the inequality constraints of the joint limits. Thus, we may get a solution of the joint variation that may violate some of the constraints. In Section 4.2 we will explain why this is not a problem if some joint limits are slightly violated and that we still can ensure plausible spine shapes. The prioritized solution is re-evaluated as long as no additional inequality constraint is violated (Figure 3.1). This additional loop is necessary to guarantee the tasks' error minimization. The cost of the clamping loop is linear to the number of recruited joints (see Section 3.2). In the worst case each clamping iteration would handle a single clamped joint. This is seldom the case, as very often multiple joints violate their limit simultaneously which is handled through a single clamping iteration.

3.2 Joint Recruiting Level

Our PIK solver allows a task to recruit all or part of the joints from its parent up to the root of the articulated structure. We can manually discard joints that should not participate to achieve the task. For example, whenever controlling the position of the wrist it is important to decide whether the spine should participate or not. Joints can be recruited by several tasks that are possibly conflicting. The problem of overlapping joint regions has first been described by [BOK80]. Normally, to resolve this problem a minimal joint recruiting rule for joints shared by multiple tasks has to be enforced [CB04]. This rule concerns those parts of the articulated structure where multiple tasks may recruit part of their joints. Let T_i be a task of priority i .

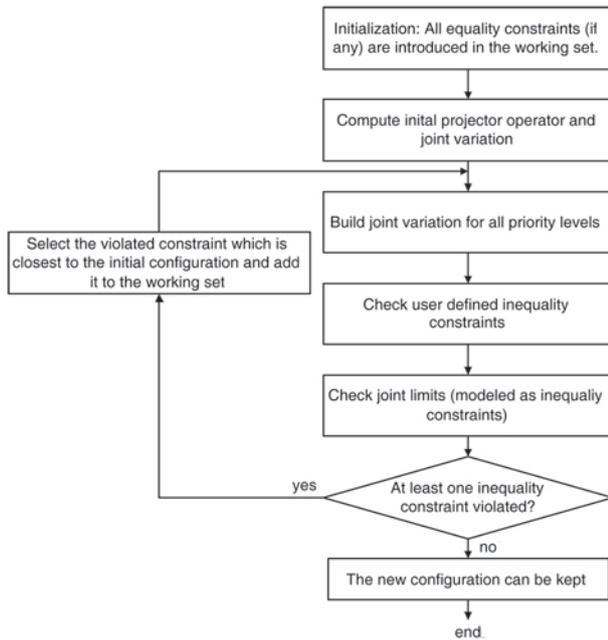


Figure 3: The Prioritized Inverse Kinematics convergence loop highlighting the construction of the joint variation solution for multiple priority levels and the management of the equality and inequality constraints.

The set of recruited joints by the task T_i is defined as $Rec(T_i)$. The set of all possible recruited joints by a task T_i is given by $Anc(T_i)$ (from its parent up to the root). Then, for any two priority levels $a > b$ we must have:

$$Rec(T_b) \cap Anc(T_a) \subset Rec(T_a) \quad (3)$$

Equation 3 states that recruited joints of low priority tasks are a subset of high priority tasks. Without this rule, it may lead to diverging solutions where a low priority task dominates a higher priority task. Equation 3 solves the problem of overlapping regions only if there are no joints coupled by (in)equality constraints. The first observation that can be made is that for joint coupling we do not have longer this simple formulation of the recruited joints. If a joint is recruited by a task T_i and if the same joint is also coupled to other joints, all these coupled joints are implicitly recruited by the task T_i . We define the set of all implicitly recruited joints due to coupling of a task T_i as $Coup(Rec(T_i))$. This can lead to the following problem: a joint recruited by a task T_b can be coupled to another joint that is only recruited by a task T_a but not recruited by T_b (with priority levels $a > b$). Thus, the task T_b would gain influence over task T_a as the equality constraints have “infinite” priority (see

section 3.1). Another problem is that joints can be coupled that are not directly recruited by any task. To solve the above mentioned problems, we extend Equation 3. For any two tasks T_a and T_b with priority levels $a > b$, we must have:

$$(Rec(T_b) \cup Coup(Rec(T_b))) \cap Anc(T_a) \quad (4)$$

$$\bigcup Coup(Rec(T_i)) \subset \bigcup Rec(T_i) \quad (5)$$

Equation 4 ensures that coupling can only take place from the joints up to the root of the articulated structure and that a lower priority task cannot couple a joint that is nearer to the root than a higher priority task. Equation 5 states that only joints recruited by tasks are allowed to be also coupled by constraints.

4 Spine

4.1 Introduction

The human spine consists of 24 movable vertebrae. According to the positions and the functionality of the individual vertebrae, the spine can be divided into three regions [Kap87][Kap82][MB91] (see Figure 4):

- Cervical region: seven vertebrae in the neck
- Thoracic region: twelve vertebrae in the thorax
- Lumbar region: five vertebrae in the abdomen

Each vertebrae has three dofs of rotation:

- rotation: flexion-extension (forward-backward)
- rotation:lateral-bending (left-right)
- torsion (axial rotation around the vertical spine axis)

These three rotational components may have quite different rotation centers and non-orthogonal rotation axes. Modeling each vertebra as an individual joint without taking into consideration the coupling behavior that exists among them due to the rotational behavior (see Figure 1) is not recommended as too much freedom is left in the spine for the PIK solver. This choice usually leads to unrealistic spine postures. Instead, it is preferable to use only a few uncoupled/independent joints strategically placed on the spine, in order to have a more realistic rigidity of the system. By a simple reduction of the dofs of the spine due to coupling, the control of the spine is also simplified.

4.2 Spine Model

As already mentioned, each vertebra allows a swing motion (corresponds to flexion-extension and lateral-bending) and a torsion motion (rotation along the vertical vertebrae axis), the general solution would be to represent each vertebrae by a so called ball-and-socket joint which has 3 dofs (2 for the swing motion and 1 for the torsion). To express these segment orientation Euler angles, quaternions or the exponential map are commonly used. For our PIK solver the exponential maps are advantageous compared to Euler angles or quaternions as the exponential maps do not have a singular configuration within their mobility range [Gra98]. For this reason our inverse kinematics solver uses the exponential map. For modeling the spine coupling we need to be able to couple the individual motion components of the vertebrae by simple (in)equality constraints (compare Figure 1 where we need to couple axial rotation with lateral-bending). Such a model is not supported using ball-and-socket joints represented by the exponential map as changing the swing component (models the flexion-extension and lateral-bending) also may change the torsion of this joint too. To separately control the coupling behavior of joints we have chosen to consider two distinct joint types:

- Swing joint model (2 dofs): the swing joint model is an exponential map vector with zero contribution along the main spine axis (no torsion).
- Revolute joint model (1 dof): oriented along the spine main axis to model the torsion mobility.

The choice of the swing joint is mostly to avoid singularities in the mobility range. These two joint types are strategically placed over the spine as seen in Figure 4. Owing to this organization we obtain a great flexibility in the coupling schemes. The actual coupling of the joints is based on (in)equality constraints (see Equation 1 and Equation 2). Mostly we want only to couple two components (e.g. we may couple the torsion of a joint with the lateral-bending in the cervical spine [WP90]). We add for each of the desired coupling behavior a new equality constraint where only two entries of the vector c are non-zero. Thus, we are able to fully control the coupled behavior of the human spine by using two common joint models and simple (in)equality constraints.

The concrete coupling coefficients are based on the vertebrae mobility ranges and their real human cou-

pling behavior [Kap82][Kap87]. Figure 5 presents our choice of the joint ranges based on our strategically placed joints over the spine. As already mentioned, the human spine is divided into three regions. Between two regions there exists some limited but independent movement [Kap82][Kap87]. As in the lumbar region the torsion is equally distributed [Kap82][Kap87] and its amount is very small, we have placed only one revolute joint in this region. In the thoracic and the cervical region we have placed more revolute joints according to the corresponding bigger torsion changes (Figure 5). To reflect this independent movement between individual regions of the spine (lumbar-thoracic, thoracic-cervical), we couple two regions with inequality constraints as illustrated in Figure 2 right. This helps the regions behave more independently than with a strict coupling with equality constraints. Nevertheless, we can ensure a fluent transition between two spine regions. The drawback using inequality constraints is that they may lead to a re-evaluation of the solution. The same problem exists for the joint limits which are also modeled as inequality constraints.

The problem that may arise modeling the coupling of the spine with (in)equality constraints is that the equality constraints and the inequality constraints (user-defined between two spine regions or used for the joint limits) may be conflicting (Section 3.1). Thus, it may not be possible to achieve all (in)equality constraints at the same time. To reduce such conflicting constraints we only check the joint limits for the first joint of each spine region:

- Swing: v_{15}, v_{t12}, v_{c7}
- Twist: v_{l2}, v_{l8}, v_{c6}

The concrete coupling coefficients are inferred from the real vertebrae mobility ranges and real measured human coupling coefficients. Let us examine this by giving two examples. In our model, all vertebrae of the thoracic region have the same mobility range for lateral bending. Thus, we have to choose a coupling coefficient that is less than one. Otherwise, a joint could violate its real mobility range limit. The difference of the joint mobility ranges always defines an upper bound for the coupling coefficients. If the joint mobility ranges decrease, we have to choose a coupling coefficient that is smaller than one. If the joint range increases we can choose a coupling coefficient that is bigger than 1.0. Consequently, the *accordance* to the

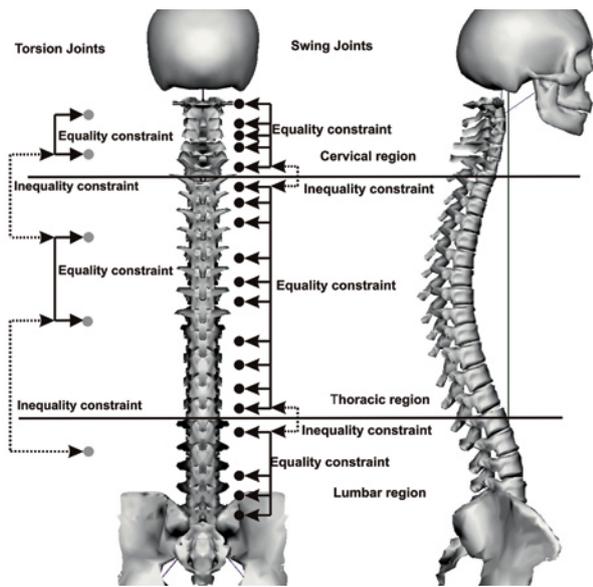


Figure 4: Our human spine model with the different spine regions. We have strategically placed some rotational joints (indicated by gray circles) to model the torsion of the spine where for the remaining joints we use swing joints (indicated by black circle). Inside a region we couple the swing joints by equality constraints (solid flashes). To keep some independency between the regions we use inequality constraints (dashed flashes)

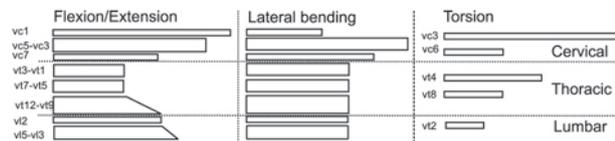


Figure 5: Joint ranges of the spine at different levels of the spine along the three anatomic axis [Kap82][Kap87].

joint limits of the joints inside a region is automatically given by the choice of the coupling coefficients that are chosen accordingly to the real vertebrae mobility ranges and real measured human coupling coefficients.

4.3 Summary

To summarize, our spine model is composed of three segments. We strictly decouple the swing and torsion motion components of the vertebrae using only swing and revolute joints. Within each segment we couple the joints by equality constraints, while the different segments are connected by inequality constraints. As already mentioned, each equality constraint removes at least one dof for the PIK solver. As we are only us-

ing equality constraints with exactly two non-zero values in the vector c for the coupling, each coupling by an equality constraint removes exactly one dof from the solution space. The thoracic region of our spine model consists of 10 swing (each 2 dof) and 2 revolute joint (each 1 dof) which gives a total of 22 dof. The simplest case would be to couple these joints using equality constraints to couple the flexion-extension with flexion-extension, the lateral bending with lateral bending and the torsion with torsion. Already in this simple example, the whole thoracic region is represented by three dofs (1 for flexion-extension, 1 for lateral-bending and 1 for torsion). The same holds for the other spine region (3 dof for each). Thus, the whole spine can be controlled by 9 dof which is considerably less than the 72 dof of the uncoupled spine. Although our model is probably still not as accurate as a real human spine, we can achieve fairly more realistic spine configurations than in an uncoupled spine. We think that our spine model is a good compromise between the accuracy and the simplicity of control.

5 Results

Three types of task controls have been exploited in the following experiments: position and/or orientation control, and projection of the position control of the center of mass on the ground (given the mass distribution of the articulated structure). The latter is especially useful for ensuring the balance of virtual mannequins.

5.1 Isolated Spine

This experiment highlights the behavior of our approach with two conflicting tasks on an isolated spine consisting of all 24 vertebrae where the vertebrae are coupled with 32 equality (inside the spine regions) and 12 inequality constraints (connection between two spine regions). Figure 6 on the left show the initial configuration of the spine. We have defined only a position task T_a (move neck slightly to the right) and an orientation task T_b (keep the orientation of the head during the motion) to be achieved with priorities $a < b$. We executed this experiment twice, once with the above described coupled spine and once where the joints of the spine can move independently within their joint mobility range.

Figure 6 highlights the initial position, the achieved for the spine without coupling and with coupling. The

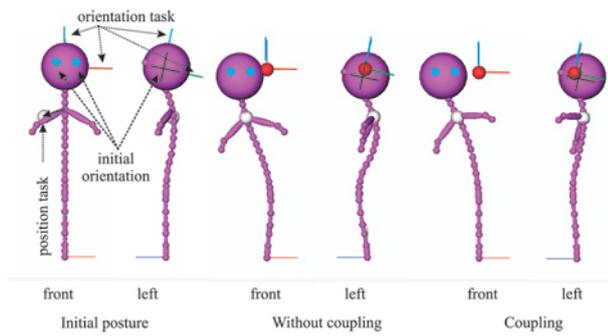


Figure 6: Front and left side views of the isolated spine: (left) initial configuration with indications of the two tasks; (middle) end configuration without coupling; (right) end configuration with coupling.

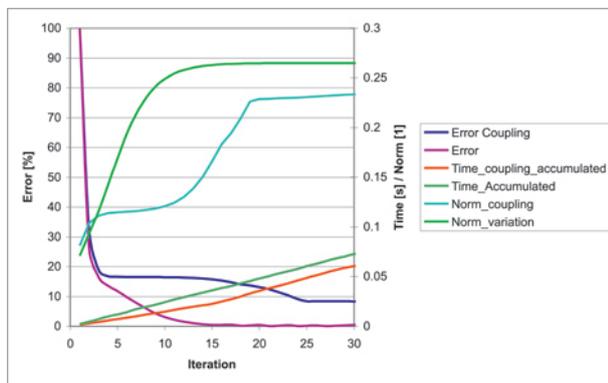


Figure 7: Comparison between coupling/without coupling for computational time and error convergence.

obvious problem of the uncoupled spine is the strong change of the spine shape while the coupled spine displays no abrupt changes of the shape. This behavior of the uncoupled spine is possible due to the lack of coupling. Each joint is able to move independently without taking into account the motions of other joints.

We have also compared the accumulated computational speed, the error convergence and the norm of the joint variations of the two methods as presented in Figure 7. Coupling with equality constraint reduces the solution space for the PIK solver. Therefore, the error convergence could be slowed down or the task may not be achievable while without coupling it would be possible. Although there is almost no visual difference, the position task for the coupled spine is not fully achieved while without coupling it is (cf. “Error” and “Error coupling” curves in Figure 7). Thus, the final error for the coupled spine does not converge to zero.

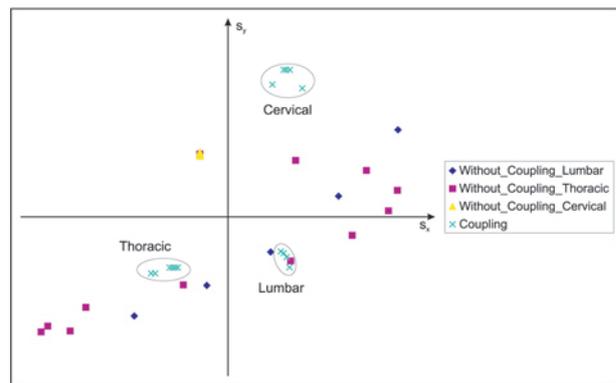


Figure 8: Distribution of the swing components of the vertebrae for coupling and without coupling.

The initial projector P_0 remains constant during the whole experiment as we are not adding or removing any equality constraint (see Section 3.1). Thus, we can pre-compute the initial projector P_0 and the additional overhead due to coupling is limited to adjusting the initial joint variation before each iteration. We have measured that around 3-8% of the total computational for one iteration of the inverse kinematics convergence process is spent to initialize the joint variation. The accumulated total computation time shows some advantages for our coupling approach. This is due to the fact that by removing dofs for the PIK solver, the solution computation becomes cheaper, the solver finds faster a solution, which compensates the additional overhead required computing the initial joint variation. Due to coupling we also restrict the overall joint variation during the convergence (cf. “Norm_variation” and “Norm_coupling” in Figure 7).

The swing components s_x (flexion-extension) and s_y (lateral-bending) are represented in Figure 8. For a real human spine it is expected that consecutive vertebrae inside a spine region have a smooth curve. But the swing component values without coupling are widely distributed in the $s_x - s_y$ space (e.g. the swing components of the thoracic region). This strong change of the swing components of the spine without coupling leads to the observed change in the spine shape as presented in Figure 6 in the middle. However, the swing component values of the vertebrae with coupling are grouped per spine region. The only significantly changes appear between two spine regions where we have coupled the vertebrae by inequality constraints to have some independent movement between the spine regions.

Another problem of the uncoupled spine is the abrupt

change of the torsion direction which is discussed in detail in Section 5.2.1.

5.2 Full Body Postures

In this section we focus on full body postures with a simplified spine representation. A full spine model consisting of 24 vertebrae may be too time consuming for real-time applications if there exist other computational expensive tasks (e.g. skinning, collision detection/response). Thus, often a compromise between full spine representation and real-time needs has to be found. In the next experiments the uncoupled simplified spine consists of 8 ball-and-socket joints with a total of 24 dofs. The lumbar and the thoracic region are each represented with three joints while the cervical region has only two joints. As already mentioned our coupled spine model can only be represented by swing and rotation joints. Our coupled spine model is built by 8 swing joints and 5 rotational joints. Figure 9 illustrates the initial posture for the next experiments.

5.2.1 Bio-mechanical considerations

In this experiment the virtual mannequin with the simplified spine has to achieve a simple balanced posture. Thus, the toes of the right foot are attracted toward a position in the back while the hands have to reach a position in the front. These goals are modeled as a middle priority position task. During the whole motion the left foot has to stay on the ground which is reflected by two high priority tasks (one for the toes and one for the heel). A low priority orientation task is used to maintain the head looking forward. To keep the virtual mannequin in balance we project the center of mass over the left foot with the highest priority task. The recruiting level is for all task set to the maximum which means up to the root of the articulated structure. The root is located just after the end of the last vertebrae of the spine. We executed this experiment twice, once with our coupling approach and once with independent vertebrae Figure 10.

Although both methods are visually similar there is an important difference. Considering three successive vertebrae Figure 10 left. The lowest vertebrae may have a torsion to the right, the middle to the left while the third vertebrae rotates again to the right. We have measured a similar behavior of the vertebrae torsion directions for the uncoupled spine as illustrated in Figure 10 middle. Such a behavior is bio-mechanically

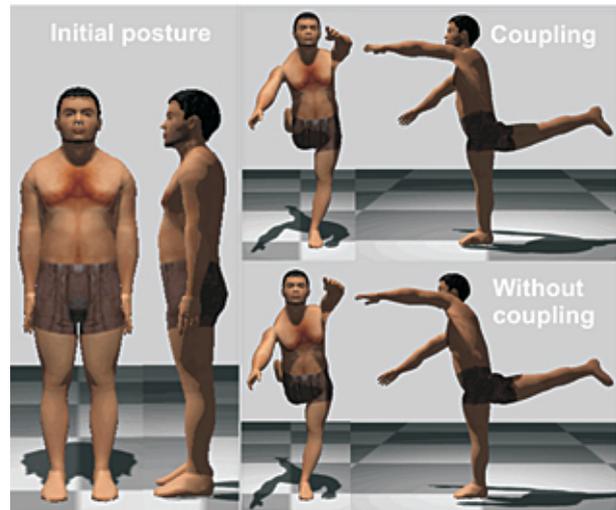


Figure 9: The initial and the achieved postures without coupling and with coupling

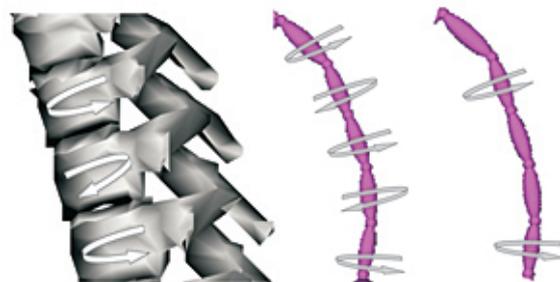


Figure 10: Possible changes of the torsion direction of vertebrae for an uncoupled spine (left) and corresponding measured values for uncoupled spine (middle) and coupled spine (right).

not possible. Changing the direction of the torsion is only possible between spine regions but not at each individual vertebra. For the coupled spine, the torsion can only change its direction for vertebrae which are coupled by inequality constraints (e.g. between two spine regions).

5.2.2 Periodic Tasks

This experiment highlights the behavior of coupling for long periodic tasks. The hands of the virtual mannequin have to follow a moving goal which forms an eight in the vertical plane (two low priority positional tasks are used). The feet have to stay on the ground (four high priority tasks) and the virtual mannequin has to keep its balance by projecting the center of mass between the two feet (highest priority task). Figure 11 illustrates the different postures without coupling

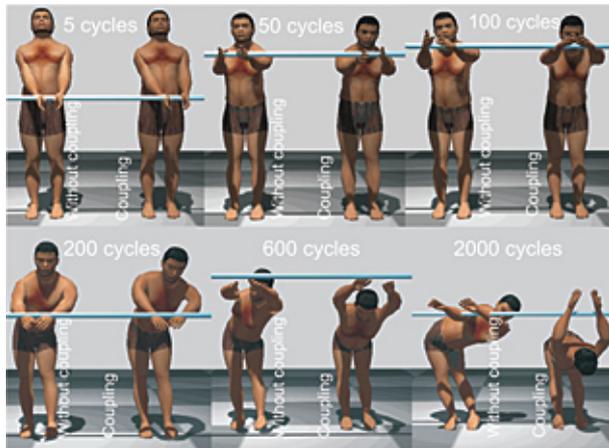


Figure 11: Postures on the left without coupling and on the right with coupling after an increasing number of cycles (from left to right and top to bottom). The blue bar indicates the goal of the wrist.

and with coupling after an increasing number of cycles. At the beginning the two postures are almost identical. We can observe that the differences between the postures increase with the number of cycles. After 2000 cycles the coupled spine has still a plausible shape while the uncoupled is highly deformed. This is caused by the drift of the solution in the joint space as originally discussed by [KH83]. Coupling joints by (in)equality constraints reduces the solution space for the inverse kinematics solver which counteracts this drift in the joint solution space for periodic tasks.

5.2.3 Real-Time Motion Capture

This experiments presents how our spine coupling can be integrated into a motion capture system for real-time IK. The user can interact with a virtual environment using a motion capture system where the user is equipped with a set of active markers (LED) and a head-mounted display (only for first person view tests) as seen in Figure 12. Individual markers provide position information only. By forming groups of two or three markers, we are able to infer also the orientation of some body parts like the head, the wrists and the toes. This allows us to define positions and/or orientation tasks of the kinematic constraints guiding the avatar posture over time. For simplicity we use the same spine model as described above. We only adapted the segment size such that they correspond the actor's segments.

The critical part is that we have ensured that the tasks that correspond to the markers are fast achieved as

there may be other time consuming tasks during the motion capture session (e.g. visualization of the environment, the virtual mannequin, obstacle avoidance). Otherwise, we would have a displacement between the user posture and the virtual mannequin posture. If this displacement becomes too big the user would not longer feel controlling the virtual mannequin. Our spine coupling has two advantages for this scenario. By reducing the solution space we would have less computational costs (see Section 5.1 and by constraining the spine shape we can use less markers. The importance to constrain the solution space to feasible spine shapes depends on the number of markers. In this example, the root position and orientation as well as the shoulder positions are constrained. These constraints already reduce the solution space of the spine. Thereby, spine coupling is not as important as using only a few markers.

6 Conclusions and Future Work

In this paper we have proposed a simple method to model the coupling behavior of the spine. We use simple (in)equality constraints to reflect the coupling behavior between the vertebrae. Our choice of equality constraints removes one dof of the solution space for the inverse kinematics solver. A relatively few parameters are sufficient to represent the whole spine (9 dof). We have shown that our approach is able to produce more natural spine shapes. Introducing constraints for the spine coupling reduces the solution space of the inverse kinematics solver. Thus, the inverse kinematics solver may find only find a solution with a higher error compared to a inverse kinematics solver without using constraints. The coupling may be too restrictive and not cover the whole space of possible human spine motions but it provides always plausible postures of the spine and is less to the drift of the solution for periodic tasks. Besides the computational costs are slightly less than for the uncoupled one as long as the equality constraints remain often constant. Otherwise, we would have to re-compute the initial projector.

Future work includes extending the method to not only couple the spine but also other structures in the human body where strong coupling behavior exists (e.g. the spine is coupled to the sacroiliac). Our actual spine model does not support different types of mobility behavior (e.g. less mobility for elderly people). We also would like to apply our coupled spine model for real-time, interactive control of virtual mannequins in a vir-

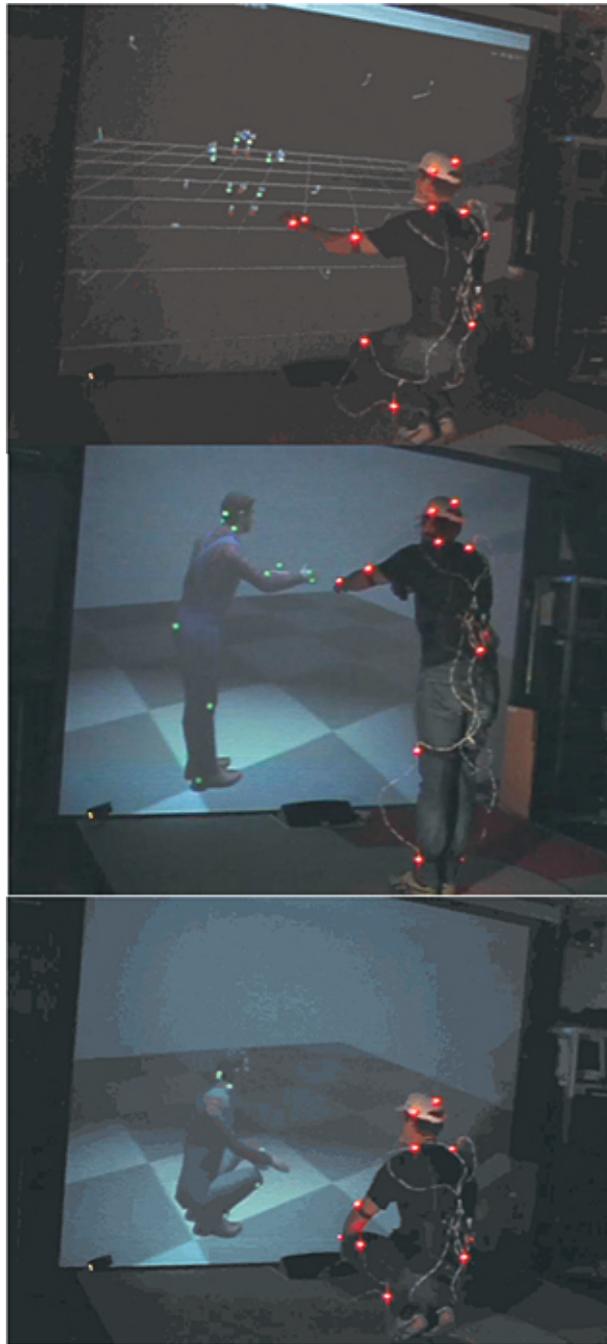


Figure 12: Interactive real-time IK with a coupled spine. Top) Marker Configuration. Middle) The user performs a reaching task. The green dots on the virtual mannequin indicate the target position of the joints. Down) Result from motion capture. The sensors (red dots) attached on the user represent the tasks for the virtual mannequin.

tual environment with obstacle avoidance where we can profit from the lower computational time of the coupled spine.

7 Acknowledgments

The authors would like to thank Achille Peternier for his support of the graphics library. The authors would also thank Damien Maupu supporting us using the graphics library and for his exploration and final set-up of the motion capture system. This work has been realized with the support of the Swiss National Science Foundation under the grants 200020-109989, and is partially supported by the E.U. ENACTIVE Network of Excellence.

References

- [BB04] Paolo Baerlocher and Ronan Boulic, *An inverse kinematics architecture enforcing an arbitrary number of strict priority levels*, *Vis. Comput.* **20** (2004), no. 6, 402–417, ISSN 0178-2789.
- [BOK80] Norman I. Badler, Joseph O'Rourke, and Bruce Kaufman, *Special problems in human movement simulation*, *Computer Graphics (SIGGRAPH '80 Proceedings)*, vol. 14, 1980, ISSN 0097-8930, pp. 189–197.
- [CB04] Benot Le Callennec and Ronan Boulic, *Interactive Motion Deformation with Prioritized Constraints*, *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, 2004*, ISSN 1727-5288, pp. 163–171.
- [DL95] S. L. Delp and P. Loan, *A graphics-based software system to develop and analyze models of musculoskeletal structures*, *Computers in Biology and Medicine* **25** (1995), no. 1, 21–34, ISSN 0010-4825.
- [GM98] Michael Girard and A. A. Maciejewski, *Computational modeling for the computer animation of legged figures*, pp. 255–262, ACM Press, New York, NY, USA, 1998, ISBN 1-58113-052-X.

- [Gra98] F. Sebastian Grassia, *Practical Parameterization of Rotations Using the Exponential Map*, Journal of Graphics Tools **3** (1998), no. 3, 29–48, ISSN 1086-7651.
- [HA99] H-Anim, *humanoid animation working group*, www.h-anim.org, 1999, Last visited November 18th, 2008.
- [HOT98] Jessica K. Hodgins, James F. O'Brien, and Jack Tumblin, *Perception of Human Motion With Different Geometric Models*, IEEE Transactions on Visualization and Computer Graphics **4** (1998), no. 4, 307–316, ISSN 1077-2626.
- [HUF05] L. Herda, R. Urtasun, and P. Fua, *Hierarchical implicit surface joint limits for human body tracking*, Comput. Vis. Image Underst. **99** (2005), no. 2, 189–209, ISSN 1077-3142.
- [Kap82] I. A. Kapandji, *The physiology of the joints : annotated diagrams of the mechanics of the human joints, Volume 1: Upper Limb*, 2nd ed. completely rev. ed., Churchill Livingstone, 1982, ISBN 0443025045.
- [Kap87] I. A. Kapandji, *The physiology of the joints : annotated diagrams of the mechanics of the human joints, Volume 3: The Trunk and the Vertebral Column*, Churchill Livingstone, 1987, ISBN 0443012091.
- [KH83] Charles A. Klein and C. H. Huang, *Review of pseudoinverse control for use with kinematically redundant manipulators*, IEEE Transactions on Systems, Man, and Cybernetics **SMC-13** (1983), no. 2, 245–250, ISSN 0018-9472.
- [KMA05] R. Kulpa, F. Multon, and B. Arnaudi, *Morphology-independent representation of motions for interactive human-like animation*, Computer Graphics Forum **24** (2005), no. 3, 343–351, ISSN 0167-7055.
- [Kor85] James Urey Korein, *A Geometric Investigation of Reach*, MIT Press, Cambridge, 1985, ISBN 0-262-11104-7.
- [MB91] Gary Monheit and Norman I. Badler, *A kinematic model of the human spine and torso*, IEEE Computer Graphics and Applications **11** (1991), no. 2, 29–38, ISSN 0272-1716.
- [MNF02] Anderson Maciel, Luciana Porcher Nedel, and Carla Maria Dal Sasso Freitas, *Anatomy-Based Joint Models for Virtual Human Skeletons*, Proceedings of computer animation CA 2002, IEEE Computer Society, 2002, ISBN 0-7695-1594-0, pp. 220–224.
- [MT00] Walter Maurel and Daniel Thalmann, *Human shoulder modeling including scapulo-thoracic constraint and joint sinus cones*, Computers & Graphics **24** (2000), no. 2, 203–218, ISSN 0097-8493.
- [MTT04] Nadia Magnenat-Thalmann and Daniel Thalmann, *Handbook of Virtual Humans*, John Wiley & Sons, 2004, ISBN 0470023163.
- [PZB90] Cary B. Phillips, Jianmin Zhao, and Norman I. Badler, *Interactive Real-Time Articulated Figure Manipulation Using Multiple Kinematic Constraints*, Proceedings of the Symposium on Interactive 3D Graphics (New York, NY, USA), vol. 24, Computer Graphics, no. 2, ACM Press, 1990, ISBN 0-89791-351-5, pp. 245–250.
- [SNTH03] Wei Shao and Victor Ng-Thow-Hing, *A general joint component framework for realistic articulation in human characters*, Proceedings of the 2003 symposium on Interactive 3D graphics SI3D, 2003, ISBN 1-58113-645-5, pp. 11–18.
- [SUS95] Mel Slater, Martin Usoh, and Anthony Steed, *Taking steps: the influence of a walking technique on presence in virtual reality*, ACM Trans. Comput.-Hum. Interact. **2** (1995), no. 3, 201–219, ISSN 1073-0516.
- [WG02] Jane Wilhelms and Allen Van Gelder, *Fast and easy reach-cone joint limits*, Journal of Graphics Tools: JGT **6** (2002), no. 2, 27–41, ISSN 1086-7651.

[WP90] Augustus A White and Manohar M Panjabi, *Clinical Biomechanics of the spine*, J.B. Lippincott Company, 1990, ISBN 0397507208.

Citation

Daniel Raunhardt and Ronan Boulic, <i>Real-Time Joint Coupling of the Spine for Inverse Kinematics</i> Journal of Virtual Reality and Broadcasting, 5(2008), no. 11, November 2008, urn:nbn:de:0009-6-15886, ISSN 1860-2037.
--