Learning Periodic Human Motion through Imitation using Eigenposes

by
Rawichote Chalodhorn Ph.D.

Humanoid Robotics Lab, Neural System Group,
Dept. of Computer Science & Engineering, University of Washington.
The ultimate goal is to learn a complex task by imitation.
Q: Can we just replay the motion?
A: Apparently not!
Problem No.1

The motion pattern needs to be optimized to match the dynamics of the robot.

But! Direct optimization of full-body "high-dimensional" joint angle data is "intractable".
Problem No.2
We bought a commercial robot, but the company just simply doesn’t give us the dynamic model. What should we do?
The dynamic model “is not” available!
Research statement

The research goal is to “generate full-body humanoid motions” while the problem of “intractable of high dimensional data” is inherited and the problem of “absences of dynamic model” is presence.
Proposed framework

Kinematic Mapping

Dimension Reduction

Inverse Mapping

Model Predictor

Action Selection (Optimization)

Robot

High dimensional data

Low dimensional data

Optimized Low dimensional data

Predicted sensory data

Actual sensory feedback

Low dimensional data
Presentation outline

- Low-dimensional subspaces
- Motion optimization algorithm
- Motion optimization results
- Motion imitation
- Lossless motion imitation
Dimension reduction algorithms

- **Linear Principal components analysis (PCA)**
  [Karhunen and Loève 1940s']

- **None-Linear PCA**
  [Kirby and Miranda, 1996]

- **Locally Linear Embedding (LLE)**
  [Roweis and Saul, 2000]

- **ISOMAP**
  [Tenenbaum et al., 2000]

- **Gaussian Process Latent Variable Models**
  [Neil D. Lawrence 2003]
Low Dimensional posture space

[Gaussian Process Latent Variable Models]

Courtesy of Keith Grochow
The “eigenpose” space

3-D low-dimensional subspaces by linear PCA
The “eigenpose” space

3-D low-dimensional subspaces by linear PCA
Action subspace embedding

Map data to cylindrical coordinate system

\[ z_{\theta} = \frac{\sum_i (\hat{x}_i \times \hat{x}_{i+1})}{\|\sum_i (\hat{x}_i \times \hat{x}_{i+1})\|} \]

Learn 1-D representation of motion in term of motion phase angle:

\[ [r, h] = g(\varphi) \]
Presentation outline

- Low-dimensional subspaces
- Motion optimization algorithm
- Motion optimization results
- Motion imitation
- Lossless motion imitation
Optimization strategy

Gyroscope signals

Optimized motion
Learning the predictive model

\[ s_{t+1} = F(s_t, \ldots, s_{t-n}, a_t, \ldots, a_{t-n}) \]
NARX model-predictor

Nonlinear autoregressive network with exogenous inputs

Feed Forward Network

input

feedback

Time Delay

output

recurrent neural network
NARX model-predictor

Nonlinear autoregressive network with exogenous inputs

$$s_{t+1} = F(s_t, s_{t-1}, a_t, a_{t-1})$$
Gyroscope signals prediction

Gyroscope signal of X-axis

Gyroscope signal of Y-axis

Gyroscope signal of Z-axis
Predictive motion generator

\[ a^*_t = \arg\min_{a_t} \Gamma(F(s_t, \ldots, s_{t-n}, a_t, \ldots, a_{t-n})) \]
Maths details

\[ a_t^* = \arg \min_{a_t} \Gamma(F(s_t, \ldots, s_{t-n}, a_t, \ldots, a_{t-n})) \]

\[ \Gamma(\omega) = \lambda_x \omega_x^2 + \lambda_y \omega_y^2 + \lambda_z \omega_z^2 \]

\[ \chi_t^* = \arg \min_{\chi_t} \Gamma(F(\omega_t, \omega_{t-1}, \chi_t, \chi_{t-1})) \]

\[ S = \begin{bmatrix} \varphi_s \\ r_s \\ h_s \end{bmatrix} \]

\[ \begin{align*}
\varphi_{t-1} & < \varphi_s \leq \varphi_{t-1} + \epsilon_{\varphi} \\
r_a - \epsilon_r & \leq r_s \leq r_a + \epsilon_r \\
h_a - \epsilon_h & \leq h_s \leq h_a + \epsilon_h \\
0 & < \epsilon_{\varphi} < 2\pi \\
[r_a, h_a] & = g(\varphi_s)
\end{align*} \]
Presentation outline

- Low-dimensional subspaces
- Motion optimization algorithm
- Motion optimization results
- Motion imitation
- Lossless motion imitation
Motion-phase optimization

\[ [r, h] = g(\varphi) \]

\[ \varphi^*_t = \arg \min_{\varphi_t} \Gamma(F(\omega_t, \omega_{t-1}, \varphi_t, \varphi_{t-1})) \]
3-D Eigenpostes optimization result
3-D Eigenposes optimization result

Gyroscope signal of X-axis
- Original : RMS = 0.3236
- Optimized : RMS = 0.0521

Gyroscope signal of Y-axis
- Original : RMS = 0.4509
- Optimized : RMS = 0.0501

Gyroscope signal of Z-axis
- Original : RMS = 0.3795
- Optimized : RMS = 0.0533
3-D Eigenposes optimization result
Presentation outline

- Low-dimensional subspaces
- Motion optimization algorithm
- Motion optimization results
- **Motion imitation**
- Lossless motion imitation
Human motion capture mapping

Human skeleton

Robot skeleton
Joint trajectories

HOAP2: joint trajectories of right leg

HOAP2: joint trajectories of left leg
Action subspace scaling

Normalized joint data
mean = 0
standard deviation = 1
Imitate a human walking gait
Imitate a human walking gait
Walking by imitation results
Walking by imitation results
Presentation outline

- Low-dimensional subspaces
- Motion optimization algorithm
- Motion optimization results
- Motion imitation
- Lossless motion imitation
Human sidestep motion
Accuracy of 3-D eigenposes

Sidestep 3-D eigenposes

Accuracy accumulation along the principal axes

81.38%  98%  100%
Hyperdimensional cylindrical transformation

For $f \in \mathbb{R}^n$ when $n > 3$

\[
\begin{align*}
  f(d_1, d_2, d_3, \ldots, d_n) \\
  f(x, y, z_1, \ldots, z_{n-2})
\end{align*}
\]

Suppose $f \in \mathbb{R}^5$

\[
\begin{align*}
  f(x, y, z_1, z_2, z_3) \\
  f(x, y, z_1) & \rightarrow f(\varphi, r, h_1) \\
  f(x, y, z_2) & \rightarrow f(\varphi, r, h_2) \\
  f(x, y, z_3) & \rightarrow f(\varphi, r, h_3)
\end{align*}
\]

Thus

\[
\begin{align*}
  f(x, y, z_1, \ldots, z_{n-2}) & \rightarrow f(\varphi, r, h_1, \ldots, h_{n-2})
\end{align*}
\]
Multiple cylindrical frames

\[ f(\varphi, r, h_1) \]
\[ f(\varphi, r, h_2) \]
\[ f(\varphi, r, h_3) \]
\[ f(\varphi, r, h_4) \]
Hyperdimensional motion optimization

Hyperdimensional action subspace embedding

\[ [r, h_1, h_2, \ldots, h_{18}] = g(\varphi) \]

Motion-phase optimization

\[ \varphi_t^* = \arg \min_{\varphi_t} \Gamma(F(\omega_t, \omega_{t-1}, \varphi_t, \varphi_{t-1})) \]
Hyperdimensional optimization result

Original posture
Optimized posture

Tuesday, February 23, 2010
Conclusion

- Stable humanoid motion can be realized through imitation
- Compact low-dimensional spaces allows efficient optimization
- Dynamic model is not required
Conclusion

- Stable humanoid motion can be realized through imitation
- Compact low-dimensional spaces allows efficient optimization
- Dynamic model is not required

Note:
- Learn directly from the real robot
- Learn none-periodic motion
- Real-time feedback needs to be realized
- Multiple learning modules organization
Last but not least
Last but not least
Last but not least
Thank you!