A Tribute to alq — Control of a Virtual Skydiver

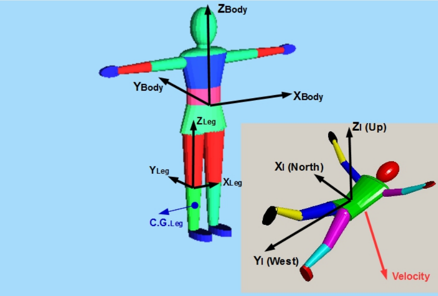
Anna Clarke and Per-Olof Gutman
Technion — Israel Institute of Technology

November 17, 2017

Outline

Coordinate Systems

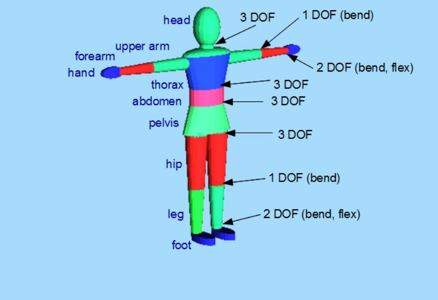
- Inertial Frame: North, West, Up
- Body Frame
- Wind Frame: transformation from Body to Wind frame includes two Euler rotations α about X-axis, and then $-\beta$ about Y-axis
- Limb Local Frame
- Local Wind Frame: transformation from Local Limb to Wind frame defines local angle of attack, sideslip, and roll



BSP (Body Segments Parameters) Equations

The body is constructed by 16 segments connected with joints, which provide in total 33 Degrees-of-Freedom. The following data is computed for each segment i:

- Principal Moments of Inertia $(Ixx_i, Iyy_i, Izz_i)_{local}$
- center of gravity (in local coordinates) $\vec{R}_{cg_i}\, local$
- origin of local coordinates relative to the parent segment it is attached $\vec{D}_{Child\ Limb_{IN\ Parent\ Limb}}$
- transformation between child and parent segments, defined by Euler angles and interpreted as a rotation quaternion $q^{Parent\ Limb}_{Child\ Limb}$



References

1. Kwon, Y.-H. (1994). KWON3D Motion Analysis Package 2.1 User's Reference Manual. Anyang, Korea: V-TEK Corporation.
2. Kwon, Y.-H. (1993). The effects of body segment parameter estimation on the experimental simulation of a complex airborne movement. Doctoral Dissertation, Pennsylvania State University.

Transformation Chain

Computing origin, center of gravity, and transformation of Local **Limb** *i* frame relative to **Body** Frame:

$$\vec{D}_{Limb_i\,IN\,Body},\ \vec{R}_{cg_i\,Body},\ q^{Limb_i}_{Body}$$

Example: Right Hand

$$q^{Right\ Hand}_{Body} = \underbrace{(q^{Right\ Forearm}_{Right\ Hand})^*}_{2\ \ \mathrm{DOF}} \otimes \underbrace{(q^{Right\ Upperarm}_{Right\ Forearm})^*}_{1\ \ \mathrm{DOF}} \otimes \underbrace{(q^{Thorax}_{Right\ Upperarm})^*}_{3\ \ \mathrm{DOF}} \otimes \underbrace{(q^{Abdomen}_{Thorax})^*}_{3\ \ \mathrm{DOF}} \otimes \underbrace{(q^{Body}_{Abdomen})^*}_{3\ \ \mathrm{DOF}}$$

$$\vec{D}_{Right\ Hand\,IN\,Body} = (q^{Body}_{Abdomen})^* \otimes (\vec{D}_{Thorax\,IN\,Abdomen} + (q^{Abdomen}_{Thorax})^* \otimes (\vec{D}_{Right\ Upperarm\,IN\,Thorax} +$$
$$+(q^{Thorax}_{Right\ Upperarm})^* \otimes (\vec{D}_{Right\ Forearm\,IN\,Right\ Upperarm} + (q^{Right\ Upperarm}_{Right\ Forearm})^* \otimes \vec{D}_{Right\ Hand\,IN\,Right\ Forearm})))$$

$$\vec{R}_{cg_{Right\ Hand\,Body}} = \vec{D}_{Right\ Hand\,IN\,Body} + q^{Right\ Hand}_{Body} \otimes \vec{R}_{cg_{Right\ Hand\,local}}$$

Summary: Body CoG and Inertia Tensor

$$\vec{R}_{cg} = \frac{\sum_{i=1}^{N_{Limbs}} \vec{R}_{cg_i \, Body} \, m_i}{\sum_{i=1}^{N_{Limbs}} m_i}$$

$$I = \sum_{i=1}^{N_{Limbs}} \left(DCM_{Body}^{Limb_i} I_{local_i} (DCM_{Body}^{Limb_i})^T + \begin{bmatrix} \Delta Y^2 + \Delta Z^2 & -\Delta X \Delta Y & -\Delta X \Delta Z \\ -\Delta X \Delta Y & \Delta X^2 + \Delta Z^2 & -\Delta Y \Delta Z \\ -\Delta X \Delta Z & -\Delta Y \Delta Z & \Delta X^2 + \Delta Y^2 \end{bmatrix}_i m_i \right)$$

where m_i - mass of Limb i, $DCM_{Body}^{Limb_i}$ - Direction Cosine Matrix, computed from quaternion $q_{Body}^{Limb_i}$, and $\begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}_i = \begin{bmatrix} X_{cg_i} - X_{cg} \\ Y_{cg_i} - Y_{cg} \\ Z_{cg_i} - Z_{cg} \end{bmatrix}$

Equations of Motion

Forces $\vec{F} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ are obtained from linear momentum derivative:

$$\vec{p} = m\vec{V} + m\vec{\Omega} \times \vec{r_{cg}}$$

$$\vec{F} = \frac{d\vec{p}}{dt} + \vec{\Omega} \times \vec{p}$$

$$\vec{F} = m\dot{\vec{V}} + m\dot{\vec{\Omega}} \times \vec{r_{cg}} + m\vec{\Omega} \times \dot{\vec{r_{cg}}} + \vec{\Omega} \times \left(m\vec{V} + m\vec{\Omega} \times \vec{r_{cg}}\right)$$

where $\vec{V} = \begin{bmatrix} U \\ V \\ W \end{bmatrix}$ - linear velocity, $\vec{\Omega} = \begin{bmatrix} P \\ Q \\ R \end{bmatrix}$ - angular velocity.

Equations of Motion

Forces summary:

$$\frac{1}{m}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} + \begin{bmatrix} \dot{Q}z_{cg} - \dot{R}y_{cg} \\ \dot{R}x_{cg} - \dot{P}z_{cg} \\ \dot{P}y_{cg} - \dot{Q}x_{cg} \end{bmatrix} + \begin{bmatrix} Q\dot{z}_{cg} - R\dot{y}_{cg} \\ R\dot{x}_{cg} - P\dot{z}_{cg} \\ P\dot{y}_{cg} - Q\dot{x}_{cg} \end{bmatrix} +$$

$$+ \begin{bmatrix} QW - RV \\ RU - PW \\ PV - QU \end{bmatrix} + \begin{bmatrix} Q(Py_{cg} - Qx_{cg}) - R(Rx_{cg} - Pz_{cg}) \\ R(Qz_{cg} - Py_{cg}) - P(Py_{cg} - Qx_{cg}) \\ P(Rx_{cg} - Pz_{cg}) - Q(Qz_{cg} - Ry_{cg}) \end{bmatrix}$$

Equations of Motion

Moments $\vec{M} = \begin{bmatrix} L \\ M \\ N \end{bmatrix}$ are obtained from angular momentum derivative:

$$\vec{L} = I\vec{\Omega} + m\vec{r_{cg}} \times \vec{V}$$

$$\vec{M} = \frac{d\vec{L}}{dt} + \vec{\Omega} \times \vec{L} + \vec{V} \times (m\vec{\Omega} \times \vec{r_{cg}})$$

$$\vec{M} = I\dot{\vec{\Omega}} + \dot{I}\vec{\Omega} + \vec{r_{cg}} \times m\dot{\vec{V}} + \dot{\vec{r_{cg}}} \times m\vec{V} + \vec{V} \times \left(m\vec{\Omega} \times \vec{r_{cg}}\right) +$$

$$+\vec{\Omega} \times (I\vec{\Omega}) + \vec{\Omega} \times (\vec{r_{cg}} \times m\vec{V})$$

where $I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$ - inertia tensor

Equations of Motion

Moments summary:

$$L = \dot{I}_{xx}P - \dot{I}_{xy}Q - \dot{I}_{xz}R + I_{xx}\dot{P} - I_{xy}(\dot{Q} - RP) - I_{xz}(\dot{R} + PQ) -$$

$$-I_{yz}(Q^2 - R^2) + (I_{zz} - I_{yy})QR + m\dot{y}_{cg}W - m\dot{z}_{cg}V +$$

$$+my_{cg}(\dot{W} + VP - QU) + mz_{cg}(-\dot{V} + WP - RU)$$

$$M = \dot{I}_{yy}Q - \dot{I}_{yx}P - \dot{I}_{yz}R + I_{yy}\dot{Q} - I_{xy}(\dot{P} + QR) - I_{yz}(\dot{R} - PQ) -$$

$$-I_{xz}(R^2 - P^2) + (I_{xx} - I_{zz})PR + m\dot{z}_{cg}U - m\dot{x}_{cg}W +$$

$$+mx_{cg}(-\dot{W} + QU - PV) + mz_{cg}(\dot{U} + WQ - RV)$$

$$N = \dot{I}_{zz}R - \dot{I}_{xz}P - \dot{I}_{yz}Q + I_{zz}\dot{R} + I_{xz}(-\dot{P} + QR) - I_{yz}(\dot{Q} + PR) +$$

$$+I_{xy}(Q^2 - P^2) + (I_{yy} - I_{xx})PQ + m\dot{x}_{cg}V - m\dot{y}_{cg}U +$$

$$+mx_{cg}(\dot{V} + UR - PW) + my_{cg}(-\dot{U} + VR - QW)$$

Equations of Motion

Forces and Moments acting on the Body:

$$\vec{F} = \sum_{i=1}^{N_{limbs}} \vec{F_a}^{\,i} + q_{Body}^{I} \otimes \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}$$

$$\vec{M} = \sum_{i=1}^{N_{segments}} \left(\vec{r_{cg}^{i}} \times \vec{F_a}^{\,i} + \vec{M_a}^{\,i} \right) + \vec{r_{cg}} \times \left(q_{Body}^{I} \otimes \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \right)$$

where $\vec{F_a}^{\,i}, \vec{M_a}^{\,i}, \vec{r_{cg}^{i}}$ - aerodynamic force and moment acting on body limb i and its center of gravity expressed in Body coordinates,q_{Body}^{I} - rotation quaternion from Inertial to Body frame, updated as follows:

$$\dot{q}_{Body}^{I} = 0.5 \begin{bmatrix} 0 & -P & -Q & -R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix} q_{Body}^{I}$$

Kinematic Equations

Computing inertial orientation $[\psi \quad \theta \quad \phi]$ from $q^I_{Body} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$:

$$\psi = atan\frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)}$$

$$\theta = asin2(q_0q_2 - q_1q_3)$$

$$\phi = atan\frac{2(q_0q_1 + q_3q_2)}{1 - 2(q_2^2 + q_1^2)}$$

Computing Body angles of attack α and sideslip β:

$$\alpha = -atan\frac{V}{W}$$

$$\beta = -asin\frac{U}{sqrt(U^2 + V^2 + W^2)}$$

Kinematic Equations

Computing rotation quaternion from Body to Wind frame:

$$q_{Wind}^{Body} = \begin{bmatrix} cos\frac{\alpha}{2} \\ sin\frac{\alpha}{2} \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} cos\frac{-\beta}{2} \\ 0 \\ sin\frac{-\beta}{2} \\ 0 \end{bmatrix}$$

Computing rotation quaternion from Local Limb i to Wind frame and local angles of attack, sideslip, and roll:

$$q_{Wind}^{Limb_i} = q_{Body}^{Limb_i} \otimes q_{Wind}^{Body} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$$

$$\alpha_i = atan\frac{2(q_0q_3 - q_1q_2)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}$$

$$\beta_i = -asin2(q_0q_2 + q_1q_3)$$

$$\gamma_i = atan\frac{2(q_0q_3 - q_1q_2)}{q_0^2 - q_2^2 - q_3^2 + q_1^2}$$

Aerodynamic Model: Forces

Aerodynamic force acting on Limb i consists of two components: L_i perpendicular and D_i parallel to the local wind direction:

$$L_i = \left(q_{Wind}^{Body}\right)^* \otimes \left(\begin{bmatrix} cos\frac{\gamma_i}{2} \\ 0 \\ 0 \\ sin\frac{\gamma_i}{2} \end{bmatrix} \otimes \begin{bmatrix} 0.5\rho A_i \left\|\vec{V}\right\|^2 (Cl_\beta)_i \\ 0.5\rho A_i \left\|\vec{V}\right\|^2 (Cl_\alpha)_i \\ 0 \end{bmatrix} \right)$$

where A_i - limb characteristic area (local xz plane), ρ - air density, $(Cl_\alpha)_i and (Cl_\beta)_i$- aerodynamic coefficients:

$$(Cl_\alpha)_i = (Cl_\alpha)_i^{max} sin(2\alpha_i)$$

$$(Cl_\beta)_i = (Cl_\beta)_i^{max} sin(2\beta_i)$$

where $(Cl_\alpha)_i^{max}, (Cl_\beta)_i^{max}$ are assumed to be known parameters

Aerodynamic Model: Forces

$$D_i = \left(q_{Wind}^{Body}\right)^* \otimes \begin{bmatrix} 0 \\ 0 \\ 0.5\rho Area_i \left\|\vec{V}\right\|^2 Cd_i^{max} \end{bmatrix}$$

where $Area_i$ - Limb area exposed to the air flow, for belly-to-earth pose can be approximated as:

$$max(A_i^{xz}\left|cos\beta_i sin\alpha_i\right|, \quad A_i^{xy}\left|cos\beta_i cos\alpha_i\right|, \quad A_i^{yz}\left|sin\beta_i\right|)$$

and Cd_i^{max} is assumed to be a known parameter

References that inspired this model

1. Thomas T. Myers, and Chi Liang, "Free Fall Analysis and Simulation Tool (FAST)", 20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, 4 - 7 May 2009, Seattle, Washington
2. Nakashima, M.; Aoyama, A.; Omod, Y. "Development of simulation method for skydiving freefall", Engineering of Sport 5, Volume 1;2004, Vol. 1, p587
3. M. T. "PAT" WORKS, JR., "Wings of Man - The Theory of Freefall Flight", 6th AIAA Aerodynamic Decelerator and Balloon Technology Conference, Houston, TX, March 5-7, 1979

Aerodynamic Model: Moments

Aerodynamic Moment acting on a body segment which is angled relative to the air flow can be approximated as follows:

$$M_i = \left(q_{Wind}^{Body}\right)^* \otimes \left(\begin{bmatrix} cos\frac{\gamma_i}{2} \\ 0 \\ 0 \\ sin\frac{\gamma_i}{2} \end{bmatrix} \otimes \begin{bmatrix} 0.5\rho A_i \left\|\vec{V}\right\|^2 Length_i(Cm_\alpha)_i \\ 0.5\rho A_i \left\|\vec{V}\right\|^2 Length_i(Cm_\beta)_i \\ 0 \end{bmatrix} \right)$$

where $Length_i$ - limb characteristic length, $(Cm_\alpha)_i and (Cm_\beta)_i$- aerodynamic coefficients:

$$(Cm_\alpha)_i = -(Cm_\alpha)_i^{max} sin(2\alpha_i)$$

$$(Cm_\beta)_i = -(Cm_\beta)_i^{max} sin(2\beta_i)$$

where $(Cm_\alpha)_i^{max}, (Cm_\beta)_i^{max}$ are assumed to be known parameters

Aerodynamic Model: Moments

After the moments of all limbs are summarized a small damping moment is added [1].This moment occurs due to the changes in the orientation of the local wind vector with rotation rates across the parachutist:

$$\vec{M}_{damp} = -0.5\rho A \left\| \vec{V} \right\|^2 Height\,\vec{Cm}_{damp}\vec{\Omega}$$

where A - overall area exposed to the airflow, *Height* - skydiver's height, $\vec{Cm}_{damp}$ damping coefficients assumed to be known, $\vec{\Omega}$ - Body angular velocity

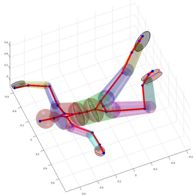
Calibration of model parameters

Skydiver dependent parameters, measured once

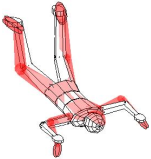
- body mass and body height
- segment dimensions
- Note: Feet are most sensitive

Aerodynamic parameters, tuned from experiments

- Air density
- drag, lift, and moment coefficients, depending on jump suit
- damping coefficients for various rotational maneuvers

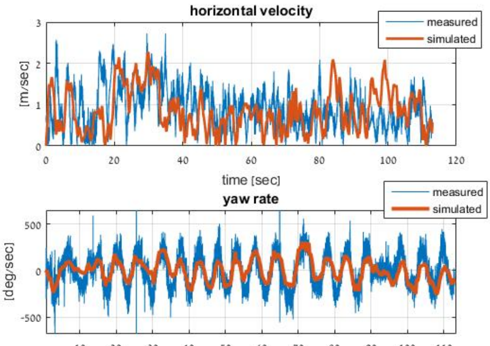


- Skydiver body model with overlaid skeleton from Xsens measurement system with 23 gyros and accelerometers under jump suit

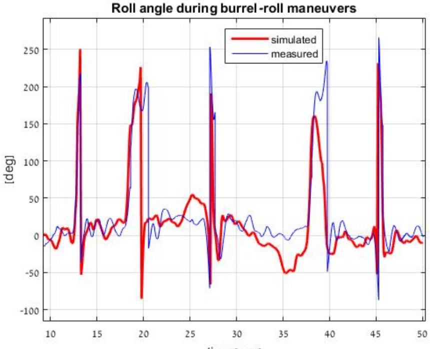


- Model pose and Xsens pose

Wind tunnel experiment: turning

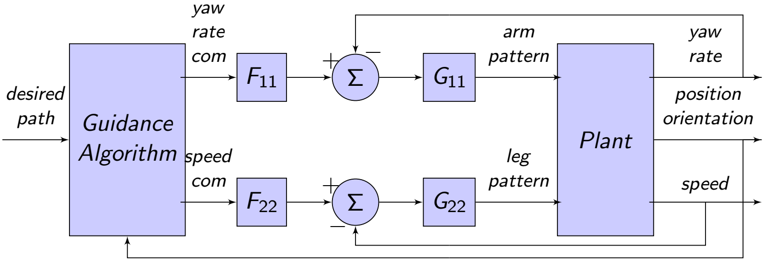


Free fall experiment: barrel roll



TITO Controller Design

The project objective - explore a possibility to design a decentralized controller able to track linear and angular velocity commands by the means of only two inputs to the non-linear skydiver plant: first associated with bending the legs (*leg pattern*), second - with deflecting the arms (*arm pattern*). A successful design is such that allows the skydiver follow a desired path, translated by a Guidance Algorithm into velocity profiles.



Block diagram: TITO decentralized design (precompensator $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, will be modified during the following design stages)

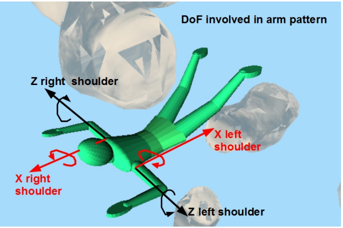
Design Degrees of Freedom

The following decisions are made during the design process:

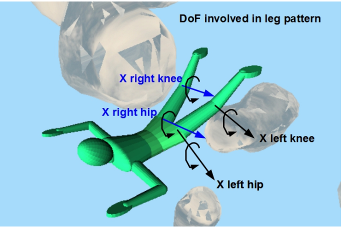
- **Coordinate Systems:** reference signals (speed and yaw rate) can be given in the Inertial or Body coordinate system, both options will be explored at different design stages
- **Arm Movement Pattern:** two options will be explored: a pattern normally performed by novice skydivers, and a pattern observed in experienced skydivers. The later produces very fast turns, while the former allows only slow turns and causes stability loss if faster turns are attempted. The final design will be based on the 'experienced' pattern, but the first design step will be computed for both in order to get an insight into the difference between them
- **Leg Movement Pattern:** consists of bending the knees and dropping the thighs, while the human engineering is such that bending the knees in a belly-to-earth position automatically causes some extent of dropping the thighs. Therefore, the knees angle α will be a control input, and the thighs angle β will be computed from α according to an empirically established relation.
- **Cross-Coupling:** Arm Pattern inherently has a strong coupling with the speed. Leg Pattern coupling with yaw rate will be generated by introducing some unsymmetry to the movement of right and left leg (one leg slightly lags behind the other). This effect naturally happens in skydiving, since a jumper can not see his legs and there is a natural unsymmetry between the legs and the muscle memory associated with right and left leg movement.

Movement Patterns

Arm Pattern - responsible for turning right, left - defined by combination of 4 angles: rotation about $[X_{left}, X_{right}, Z_{left}, Z_{right}]_{shoulder} = [-\alpha, \alpha, \alpha, \alpha]$, where α is the control input $u_1(t)$

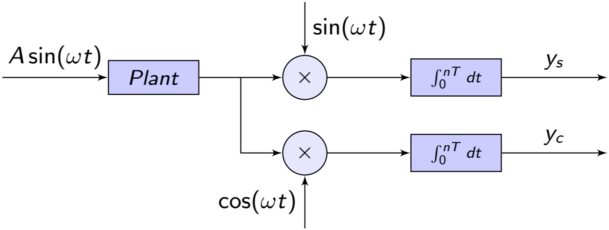


Leg Pattern - responsible for moving forwards, backwards - defined by combination of 4 angles: $[X_{left}, X_{right}]_{knee} = [\alpha, \alpha]$, $[X_{left}, X_{right}]_{hip} = [\beta, \beta]$, where α is the control input $u_2(t)$ and $\beta = \frac{\pi}{180}(-0.0035(\alpha\frac{180}{\pi})^2 - 0.0335(\alpha\frac{180}{\pi}) + 17)$



Identification of Linear Design Model

Fourier Integral Method: $|P(j\omega)| = \frac{2\sqrt{y_c^2+y_s^2}}{A}$ $\quad arg\,(P(j\omega)) = \arctan\left(\frac{y_c}{y_s}\right)$



$\omega = \frac{2\pi}{T}$

Record Transfer Functions:

$$P_{11}(s) = \frac{Y_1(s)}{U_1(s)} = \frac{Yaw\;\;Rate(s)}{Arm\;\;Pattern\;\;Angle(s)} \qquad P_{12}(s) = \frac{Y_1(s)}{U_2(s)} = \frac{Yaw\;\;Rate(s)}{Leg\;\;Pattern\;\;Angle(s)}$$

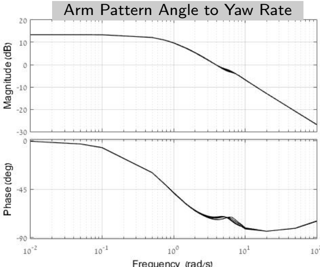
$$P_{21}(s) = \frac{Y_2(s)}{U_1(s)} = \frac{Longitudinal\;\;Velocity(s)}{Arm\;\;Pattern\;\;Angle(s)} \qquad P_{22}(s) = \frac{Y_2(s)}{U_2(s)} = \frac{Longitudinal\;\;Velocity(s)}{Leg\;\;Pattern\;\;Angle(s)}$$

Relevant frequencies and input signal amplitudes:

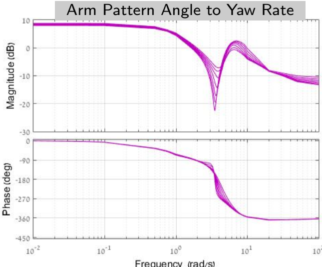
$A_{arm\;\;pattern} = [1, 3, 5, 7, 9, 11, 13, 15]\;deg$ $A_{leg\;\;pattern} = [15, 20, 25, 30]\;deg$ while each case includes the following amplitude mismatch between the right and left leg: $[0\%, \pm 15\%, \pm 25\%]$ $\omega = [0.01, 0.1, 0.25, 0.5, 0.75, 1, 1.2, 1.4, 1.5, 1.6, 1.75, 2, 2.25, 2.4, 2.5, 2.6, 2.75, 3, 3.25, 3.5, 4, 4.5, 5, 5.25, 5.5, 5.75, 6, 6.25, 6.5, 6.75, 7, 7.5, 8, 9, 10, 20, 50, 100]\;rad/sec$

Arm Patterns Comparison

Arm Pattern - Experienced Skydiver - rotation about $[X_{left}, X_{right}, Z_{left}, Z_{right}]_{shoulder} = [-\alpha, \alpha, \alpha, \alpha]$, where α is the control input $u_1(t)$

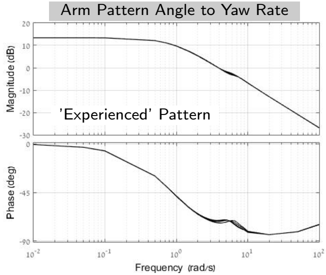


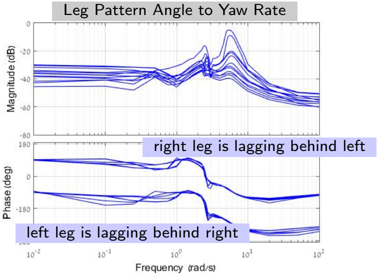
Arm Pattern - Novice Skydiver - rotation about $[X_{left}, X_{right}, Y_{left}, Y_{right}]_{shoulder} = [-\alpha, \alpha, -\alpha, \alpha]$, where α is the control input $u_1(t)$

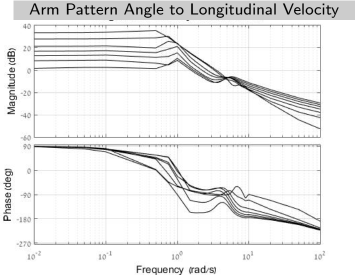


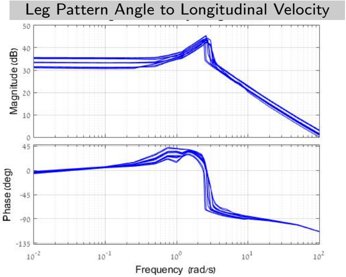
Plant Transfer Functions (in Body Coordinates): $P_{11}(s) = \frac{Yaw\ Rate(s)}{Arm\ Pattern\ Angle(s)}$

$P_{21}(s) = \frac{Longitudinal\ Velocity(s)}{Arm\ Pattern\ Angle(s)}$ $P_{12}(s) = \frac{Yaw\ Rate(s)}{Leg\ Pattern\ Angle(s)}$ and $P_{22}(s) = \frac{Longitudinal\ Velocity(s)}{Leg\ Pattern\ Angle(s)}$





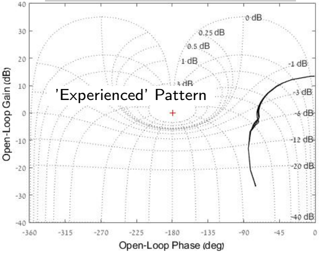




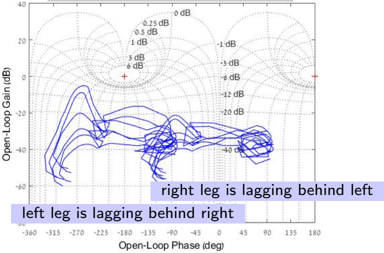
Plant Transfer Functions (in Body Coordinates): $P_{11}(s) = \frac{Yaw\ Rate(s)}{Arm\ Pattern\ Angle(s)}$

$P_{21}(s) = \frac{Longitudinal\ Velocity(s)}{Arm\ Pattern\ Angle(s)}$ $P_{12}(s) = \frac{Yaw\ Rate(s)}{Leg\ Pattern\ Angle(s)}$ and $P_{22}(s) = \frac{Longitudinal\ Velocity(s)}{Leg\ Pattern\ Angle(s)}$

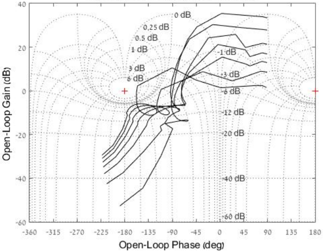
Arm Pattern Angle to Yaw Rate



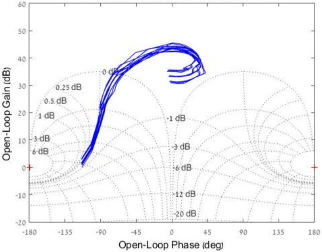
Leg Pattern Angle to Yaw Rate



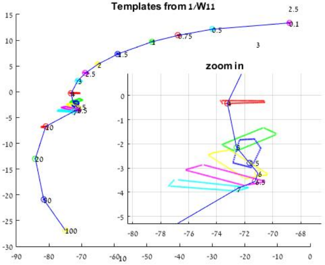
Arm Pattern Angle to Longitudinal Velocity

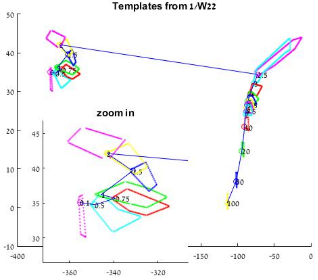


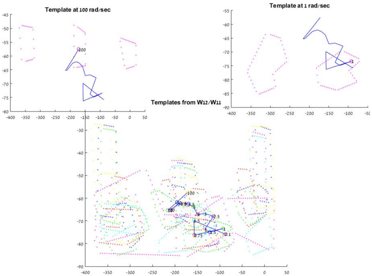
Leg Pattern Angle to Longitudinal Velocity

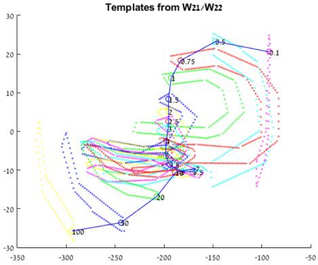


TITO Templates









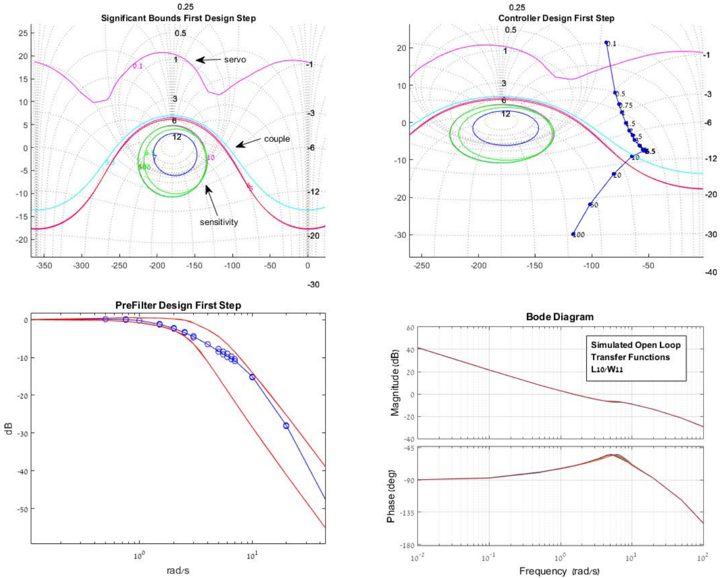
Design Specifications

- Servo Spec: rise time of between 0.5 and 1.2 seconds to the default level 50%, a maximum overshoot of 10%, a settling time of 3 seconds to within 5%
- Sensitivity Spec: 3db for the first design step, 6db for the second design step
- Cross coupling Specs: -35db for $T_{12}(s) = \frac{Y_1(s)}{R_2(s)} = \frac{Yaw\ \ Rate(s)}{Desired\ \ Speed(s)}$ and -15db for $T_{21}(s) = \frac{Y_2(s)}{R_1(s)} = \frac{Longitudinal\ \ Velocity(s)}{Desired\ \ Yaw\ \ Rate(s)}$
- Zero steady-state error

Design Strategy

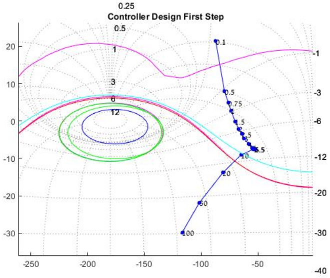
First, design the yaw rate loop (tracking desired yaw rate by the means of *arm pattern*) since the coupling observed in P_{12} is small and arrises only from inaccuracy of the leg pattern execution (syncronization between left and right leg's movement). Representing the cross-coupling by worst-case 'disturbances' will not significantly restrict the design.
Second, design the velocity loop (tracking desired speed by the means of *leg pattern*), with correct cross-coupling, since the coupling observed in P_{21} is very strong due to the nature of u_1: the arm pattern execution in addition to turning induces a backward slide. This effect is well known amongst skydivers and constitutes a serious challenge for novice jumpers.

Step 1: Arm Pattern Loop (Experienced Skydiver Pattern)

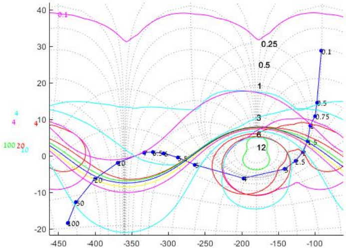


Step 1: Comparison of two Arm Patterns (Experienced vs. Novice)

Arm Pattern of an Experienced Skydiver



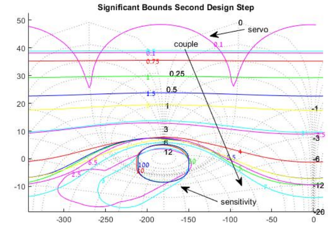
Arm Pattern of a Novice Skydiver

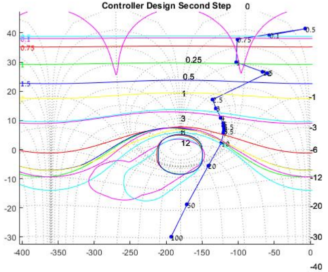


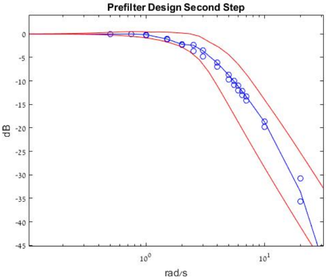
Observations

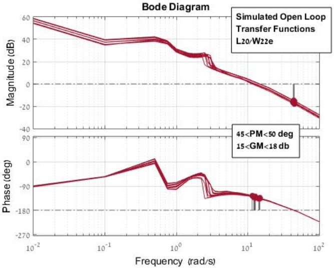
- The best stable design that uses the 'novice' pattern does not satisfy any one of the three specifications. This happens due to the anti-resonance - resonance pair observed earlier in the plant transfer functions.
- Novice skydivers perform turning very slowly and easily loose stability if attemp to speed-up their maneuvers. This fact (though often considered a phycological factor) inherently follows from the above nichols plots.
- 'Experienced' pattern is not taught to novicies due to its complexity (the body never moved in such a way, whereas the 'novice' pattern resembles swimming and is easily reproduced in the air after watching its demonstration on the ground)
- The body converges to perform an 'experienced' pattern (after a few hundred jumps), however experienced skydivers are not aware of how they perform high-performance turns, as this happens at a subconscious level.

Step 2: Leg Pattern Loop (using Experienced Arm Pattern loop)









Summary of the Controllers

The design is decentralized: $P0 = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$

Step 1:

$$L_{10} = \frac{0.25(1+\frac{s}{3.5})}{s} \frac{1+\frac{s}{0.7}}{1+\frac{s}{10}} \frac{1}{1+\frac{s}{100}}$$

$$F_{11} = \frac{1}{(1+\frac{s}{7})(1+\frac{s}{8})} \frac{1+\frac{s}{0.6}}{1+\frac{s}{1}}$$

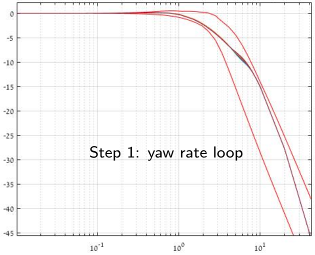
Step 2:

$$L_{20} = \frac{0.143(1+\frac{s}{3})}{s} \frac{1+\frac{s}{2}}{1+\frac{s}{6}} \frac{1}{1+\frac{s}{20}} \frac{1+\frac{2*1.5*s}{0.6}+\frac{s^2}{0.6^2}}{1+\frac{2*0.1*s}{0.6}+\frac{s^2}{0.6^2}}$$

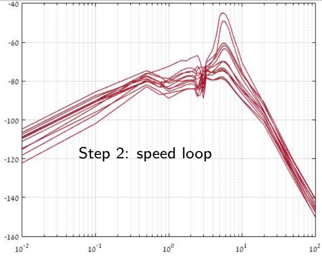
$$F_{22} = \frac{1}{(1+\frac{s}{3})^2} \frac{1+\frac{s}{1}}{1+\frac{s}{1.2}}$$

Specifications Verification in the Frequency Domain (in Body Coordinates)

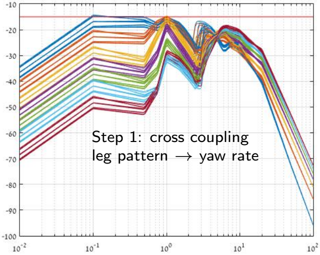
Desired Yaw Rate to Measured Yaw Rate



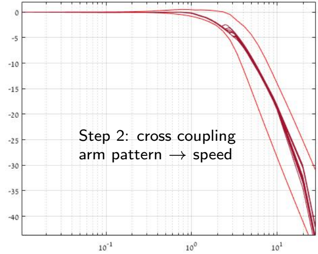
Desired Longitudinal Velocity to Measured Yaw Rate



Desired Yaw Rate to Measured Long. Velocity

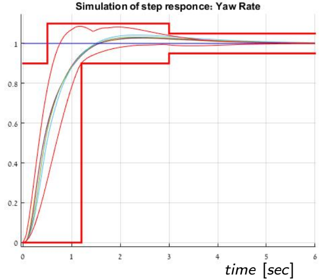


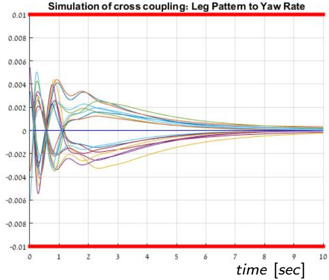
Desired Long. Velocity to Measured Long. Velocity

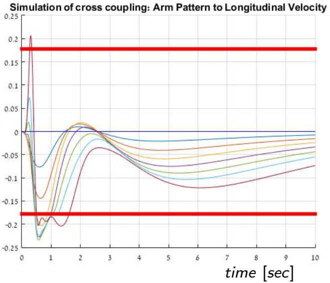


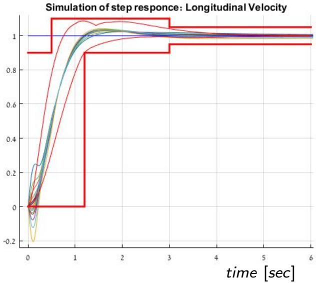
Simulations: Non-Linear Plant Implemented in Matlab

Reference Yaw rate and Speed in Body Coordinates

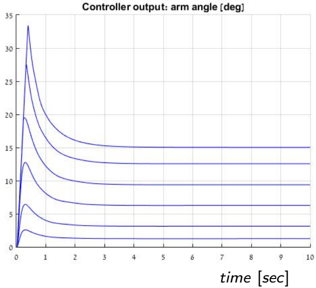


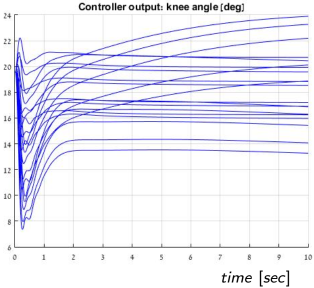


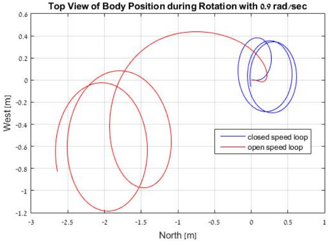




Simulations (Continued)

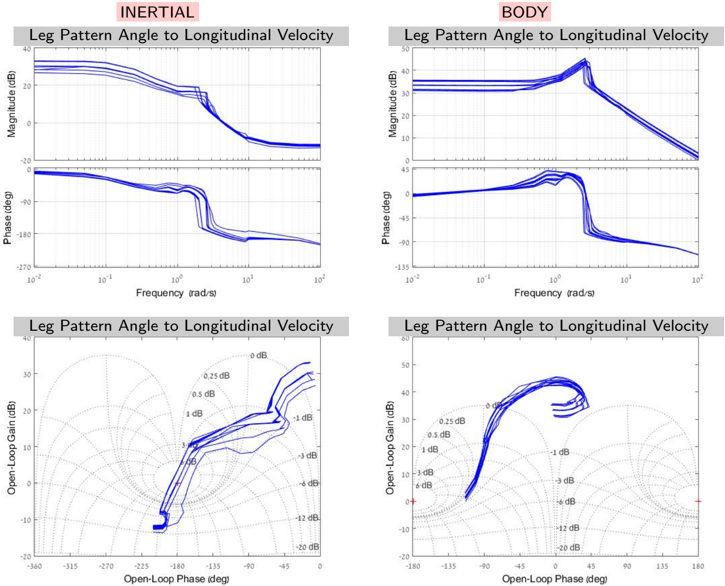






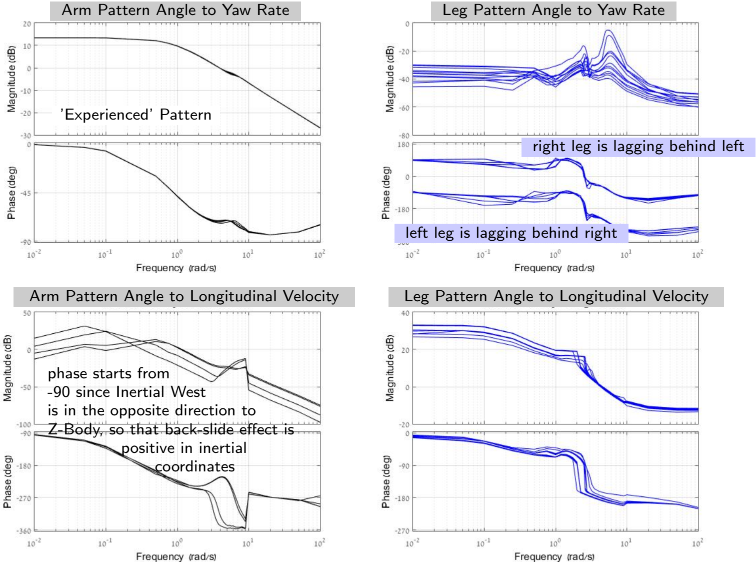
Summary: The design of both loops satisfies all specifications and is verified by the simulations of non-linear model. Nevertheless, it appears inefficient to use these loops for tracking angular and linear velocity profiles produced by a Guidance Algorithm that aims to follow a desired path. The reason is that the Guidance Algorithm produces an **Inertial** velocity command, whereas the control loop was designed to track the Body longitudinal velocity. The transformation between Body and Inertial frames is known at each simulation step, however translating the command into the Body frame gives poor performance, since the inertial velocity has different dynamics, as shown in the next section.

Longitudinal Transfer Functions: P_{22} *Inertial vs.* P_{22} *Body Frame*

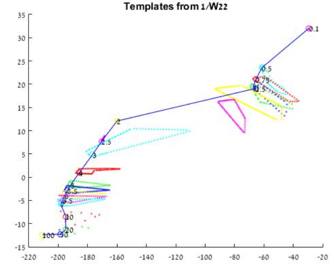


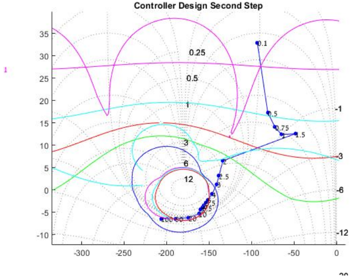
Plant Transfer Functions (in Inertial Coordinates): $P_{11}(s) = \frac{Yaw\ Rate(s)}{Arm\ Pattern\ Angle(s)}$

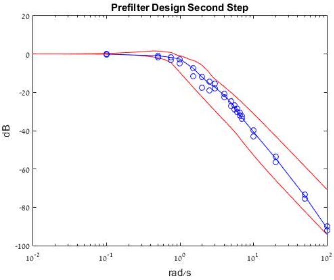
$P_{21}(s) = \frac{Longitudinal\ Velocity(s)}{Arm\ Pattern\ Angle(s)}$ $P_{12}(s) = \frac{Yaw\ Rate(s)}{Leg\ Pattern\ Angle(s)}$ and $P_{22}(s) = \frac{Longitudinal\ Velocity(s)}{Leg\ Pattern\ Angle(s)}$



Step 2 - New Design Velocity Loop in Inertial Coordinates





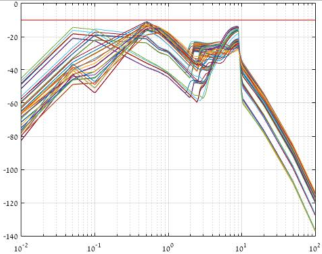


Observations:

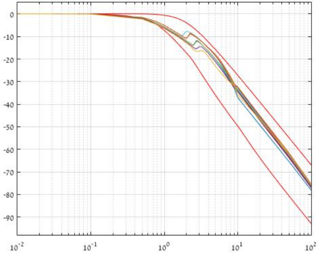
- Since the dynamics of inertial P_{22} is much slower, the specifications must be reasonably relaxed. The new servo spec: rise time of 1 - 4 sec, a settling time of 3 sec, and a maximum overshoot of 5%. The new cross-coupling spec for T_{21} is -10db.
- This difference between inertial and body longitudinal velocity is a well known issue amongst skydivers: initiating forward motion starts with pitching down thus increasing the vertical component of inertial velocity much more considerably than the horizontal component. The ability to reduce this effect as much as possible and as fast as possible is **the most crucial** skydiving skill, and it is acquired from experience.

Step 2 - New Design - Specs Verification in the Frequency Domain

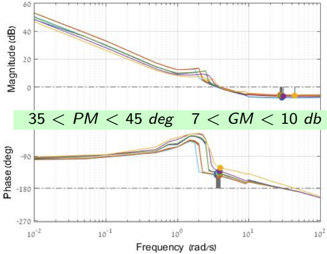
Desired Yaw Rate to Measured Long. Velocity



Desired Long. Velocity to Measured Long. Velocity



Simulated Open loop Transfer Functions L_{20}/W_{22e}



$35 < PM < 45\ deg \quad 7 < GM < 10\ db$

Design Summary:

Controller:

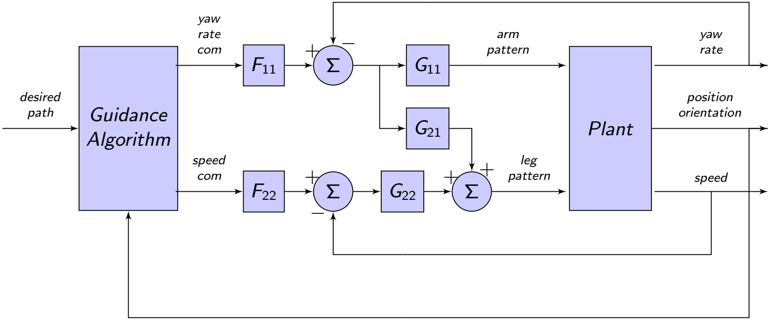
$$L_{20} = \frac{0.1(1+\frac{s}{1.5})}{s} \frac{1+\frac{s}{0.2}}{1+\frac{s}{0.6}} \frac{1+\frac{s}{1}}{1+\frac{s}{10}}$$

Prefilter:

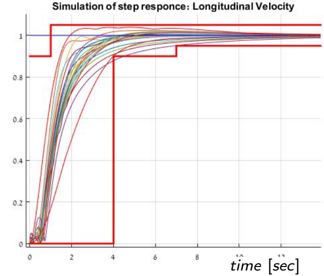
$$F_{22} = \frac{1}{(1+\frac{s}{1})(1+\frac{s}{2})}$$

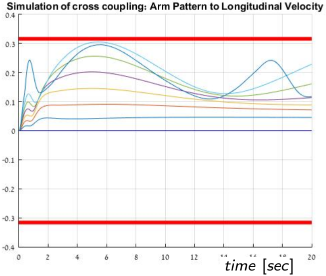
Step 2 - Pre-compensator Design

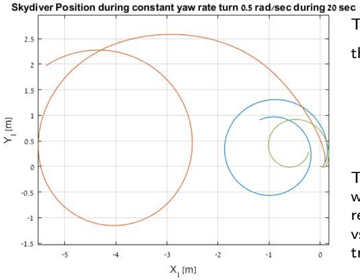
Motivation: Even though all specs were shown to be satisfied, the non-linear simulation shows that certain amplitudes of the yaw rate command cause undesirable velocity that is outside of the cross-coupling spec. This issue can be resolved if the design includes a Pre-compensator other than I ($P0 = I$ in a decentralized design shown earlier). The idea is to use the knowledge of P_{21} (a turn induces a backward slide), and to compensate for the undesirable velocity by the means of a 'faster' controller than L_{20}.



Step 2 - Simulations - Last Design Update







The Pre-Compensator $P0 = \begin{bmatrix} I & 0 \\ P0_{21} & I \end{bmatrix}$ is chosen such that the following controller is obtained:

$$G_{21} = \frac{-0.035(1+\frac{s}{3})}{s} \frac{1+\frac{s}{1}}{s} \frac{1+\frac{s}{0.5}}{1+\frac{s}{5}}$$

The cross-coupling in the non-linear simulation is now within the spec. The top view shows the improvement resulting from the new pre-compensator: green trajectory vs. blue trajectory (decentralized design), while the red trajectory is the open velocity loop.

Guidance Algorithm

The following Guidance Algorithm was used to simulate a basic task in the skydiving discipline called "Relative Work": flying towards another skydiver and stopping in front of him. This algorithm was developed for the path following task performed by autonomous vehicles, and successfuly implemented in many vehicles, working with different navigation concepts: inertial navigation, SLAM, and vision-only.

- Desired path can be any path connecting the initial position and the point of destination. The path is defined by equally spaced points along the trajectory with coordinates $(X_I, Y_I)_i, \ 1 \leq i \leq n$
- The local curvature is computed along the path (numerically, from each pair of adjucent path segments)
- The dynamic constraints are defined (max allowed speed, longitudinal and lateral acceleration)
- The velocity profile is computed: maximum velocity is computed at each path point, subject to the dynamic constraints
- The path is translated into the time domain: spline is used to create a path as a function of time according to the computed earlier velocity profile. The time vector is spaced with the simulation time step dt
- At each simulation step two points are extracted from the path vector: $X(ind_1), Y(ind_1)$ and $X(ind_2), Y(ind_2)$, where $ind_1 = \frac{lookahead\ time}{dt}$ and $ind_2 = \frac{2*(lookahead\ time)}{dt}$. The speed and yaw rate commands are computed as:

$$V_{com} = \frac{1}{lookahead\ time} \left\| (X(ind_1), Y(ind_1)) - (X_{pos}, Y_{pos}) \right\|$$

$$Yaw\ Rate_{com} = \frac{1}{lookahead\ time} \left(atan\left(\frac{X(ind_2) - X_{pos}}{Y(ind_2) - Y_{pos}} \right) - atan\left(\frac{V_x}{V_y} \right) \right)$$

where X_{pos}, Y_{pos} is the inertial position, V_x, V_y - inertial velocity, and *lookahead time* conveys the bandwidth of the tracking loops. In our case $lookahead\ time = 4\ sec$ was chosen

Simulation of Basic Relative Work (RW)

Red star - skydiver position
Green circle - lookahead point for computing the yaw rate command
Black circle - lookahead point for computing the speed command