EE 364a HW5

Yen-Yu Chang

February 15, 2020

5.17 Let $f_i(x) = \sup_{a \in P_i} a^T x$, then the problem can be expressed as:

$$\begin{array}{ll} \text{minimize} & c^T x, \\ \text{subject to} & f_i(x) \leq b_i, i = 1, \ldots, m \end{array}$$

And the optimal value of $f_i(x)$ can be expressed as a linear programming problem as:

$$\begin{array}{ll} \text{maximize} & x_i^T a, \\ \text{subject to} & C_i a \preceq d_i \end{array}$$

We can reformulate the problem as:

$$\begin{array}{ll} \text{minimize} & -x_i^T a, \\ \text{subject to} & C_i a \preceq d_i \end{array}$$

Then the *Lagrangian* is:

$$L(a, z_i) = -x_i^T a + z_i^T (C_i a - d_i)$$

The dual function is:

$$g(z_i) = \inf_a (-x_i^T a + z_i^T (C_i a - d_i))$$

Then we calculate the derivative of the *Lagrangian* over a equals to 0, we get:

$$\nabla_a L = -x_i + C_i^T z_i = 0 \implies C_i^T z_i = x_i$$

Then we can substitute $C_i^T z_i = x_i$ into the dual function to get the dual problem as:

$$\begin{array}{ll} \text{maximize} & -d_i^T z_i, \\ \text{subject to} & z_i \succeq 0, C_i^T z_i = x_i \end{array}$$

This dual problem is equivalent to:

$$\begin{array}{ll} \text{minimize} & d_i^T z_i, \\ \text{subject to} & z_i \succeq 0, C_i^T z_i = x_i \end{array}$$

The optimal value of this linear programming problem is also equal to $f_i(x)$, so we have $f_i(x) \leq b_i$ if and only if there exists a z_i such that

$$d_i^T z_i \leq b_i, \quad z_i \succeq 0, \quad C_i^T z_i = x_i$$

Hence, the original problem can be expressed as:

$$\begin{array}{ll} \text{maximize} & c^T x, \\ \text{subject to} & d_i^T z_i \leq b_i, \quad i = 1, \ldots, m \\ & C_i^T z_i = x, \quad i = 1, \ldots, m \\ & z_i \succeq 0, \quad i = 1, \ldots, m \end{array}$$

with variables $x \in \mathbf{R}^n$ and $z_i \in \mathbf{R}^{m_i}$, $i = 1, \ldots, m$.

5.40 The dual problem is:

$$\begin{array}{ll} \text{minimize} & \frac{1}{t}, \\ \text{subject to} & \sum_{i=1}^{p} x_i v_i v_i^T \succeq tI, \\ & x \succeq 0, \quad \mathbf{1}^T x = 1 \end{array}$$

The *Lagrangian* is:

$$\begin{aligned} L(t, x, \Lambda, \lambda, \nu) &= \frac{1}{t} + \mathbf{tr}\left(\Lambda(tI - \sum_{i=1}^{P} x_i v_i v_i^T)\right) - \lambda^T x + \nu(\mathbf{1}^T x - 1) \\ &= \frac{1}{t} + t\mathbf{tr}(\Lambda) + \sum_{i=1}^{P} x_i(-v_i^T \Lambda v_i - \lambda_i + \nu) - \nu \end{aligned}$$

The minimum over x_i is bounded below only if $-v_i^T \Lambda v_i - \lambda_i + \nu = 0$.
To minimize over t, we know that:

$$\inf_{t>0} \left(\frac{1}{t} + t\mathbf{tr}(\Lambda)\right) = \begin{cases} 2\sqrt{\mathbf{tr}(\Lambda)} & \text{if } \Lambda \succeq 0 \\ -\infty & \text{if } \Lambda \not\succeq 0 \end{cases}$$

Then the dual function is:

$$g(\Lambda, \lambda, \nu) = \begin{cases} 2\sqrt{\mathbf{tr}(\Lambda)} - \nu & \text{if } v_i^T \Lambda v_i + \lambda_i = \nu, \, \Lambda \succeq 0 \\ -\infty & \text{otherwise} \end{cases}$$

The dual problem is:

$$
\begin{array}{ll}
\text{maximize} & 2\sqrt{\mathbf{tr}(\Lambda)} - \nu, \\
\text{subject to} & v_i^T \Lambda v_i \leq \nu, \quad i = 1, \ldots, p, \\
& \Lambda \succeq 0
\end{array}
$$

We can define $\Lambda = \nu\Lambda'$, then the dual problem becomes:

$$
\begin{array}{ll}
\text{maximize} & 2\sqrt{\nu}\sqrt{\mathbf{tr}(\Lambda')} - \nu, \\
\text{subject to} & v_i^T \Lambda' v_i \leq 1, \quad i = 1, \ldots, p, \\
& \Lambda' \succeq 0
\end{array}
$$

And we know $\nabla_\nu \left(2\sqrt{\nu}\sqrt{\mathbf{tr}(\Lambda')} - \nu\right) = \frac{1}{\sqrt{\nu}}\sqrt{\mathbf{tr}(\Lambda')} - 1$, so optimizing over ν gives $\nu = \mathbf{tr}(\Lambda')$.

Thus, the dual problem becomes:

$$
\begin{array}{ll}
\text{maximize} & \mathbf{tr}(\Lambda'), \\
\text{subject to} & v_i^T \Lambda' v_i \leq 1, \quad i = 1, \ldots, p, \\
& \Lambda' \succeq 0
\end{array}
$$

6.3

(a) Let $y_i = \phi(a_i^T x - b_i)$. By the definition of $\phi(u)$, we have $y_i = \phi(a_i^T x - b_i) \geq 0$. It implies that $y \succeq 0$, and by the definition of $\phi(u)$, we know that $\phi(u) \geq |u| - a$, which implies $y_i = \phi(a_i^T x - b_i) \geq |a_i^T x - b_i| - a \iff y_i + a \geq a_i^T x - b_i \geq -y_i - a$. Combine all the equations and inequalities above, we can reformulate the problem as:

$$
\begin{array}{ll}
\text{minimize} & \mathbf{1}^T y, \\
\text{subject to} & -y - \mathbf{1}^T a \preceq Ax - b \preceq y + \mathbf{1}^T a, \\
& y \succeq 0
\end{array}
$$

It is a linear programming problem with variables $y \in \mathbf{R}^m, \quad x \in \mathbf{R}^n$.

(b) Let $y_i = a_i^T x - b_i$. By the definition of $\phi(u)$, we know that $\phi(y_i) \geq -a^2 \log\left(1 - (\frac{y_i}{a})^2\right)$. Then

$$
\sum_{i=1}^m \phi(a_i^T x - b_i) = \sum_{i=1}^m \phi(y_i) = \sum_{i=1}^m -a^2 \log\left(1 - (\frac{y_i}{a})^2\right) = -a^2 \log\left(\prod_{i=1}^m (1 - (\frac{y_i}{a})^2)\right)
$$

with

$$
|y_i| < a
$$

Then we know that:

$$\min. \prod_{i=1}^{m} \phi(a_i^T x - b_i) \iff \min. -a^2 \log \left(\prod_{i-1}^{m} (1 - \frac{y_i}{a})^2 \right) = \max. \prod_{i=1}^{m} \left(1 - (\frac{y_i}{a})^2\right)$$

And we can set $\left(1 - (\frac{y_i}{a})^2\right) = t_i^2$, then we can reformulate the problem as:

$$\begin{array}{ll} \text{maximize} & \prod_{i=1}^{m} t_i^2, \\ \text{subject to} & \left(1 - (\frac{y_i}{a})^2\right), \quad i = 1, \dots, m \\ & -1 \prec \frac{y_i}{a} \prec 1, \quad i = 1, \dots, m \\ & y = Ax - b \end{array}$$

with variables $t \in \mathbf{R}^m, x \in \mathbf{R}^n$, and $y \in \mathbf{R}^m$.

(c) Let $y_i + z_i = |a_i^T x - b_i|$, where $z_i \geq 0$ and $0 \leq y_i \leq M$. Then we have

$$\phi(a_i^T x - b_i) = \begin{cases} |a_i^T x - b_i|^2, & \text{if } |a_i^T x - b_i| \leq M \\ M(2|a_i^T x - b_i| - M), & \text{if } |a_i^T x - b_i| > M \end{cases}$$

If $|a_i^T x - b_i| \leq M$, we can let $y_i = |a_i^T x - b_i|$ and $z_i = 0$, so $\phi(a_i^T x - b_i) = |a_i^T x - b_i|^2 = y_i^2$. If $|a_i^T x - b_i| > M$, we can let $y_i = M$ and $z_i > 0$, so $\phi(a_i^T x - b_i) = M\left(2(M + z_i) - M\right) = M^2 + 2Mz_i$. Hence, the problem can be reformulated as:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} \left(y_i^2 + 2Mz_i\right), \\ \text{subject to} & -y - z \preceq Ax - b \preceq y + z, \\ & 0 \preceq y \preceq M\mathbf{1} \\ & z \succeq 0 \end{array}$$

(d) The original problem can be formulated as:

$$\begin{array}{ll} \text{minimize} & t, \\ \text{subject to} & \frac{1}{t} \leq \frac{a_i^T x}{b_i} \leq t, \quad i = 1, \dots, m \end{array}$$

over $x \in \mathbf{R}^n, t \in \mathbf{R}$.

The left inequality gives us the hyperbolic constraints as:

$$t a_i^T x \geq 1, \quad t \geq 0, \quad a_i^T x \geq 0$$

Hence, the problem can be reformulated as: $\begin{bmatrix} t & \sqrt{b_i} \\ \sqrt{b_i} & a_i^T x \end{bmatrix} \succeq 0$, which is a LMI constraint.

(e) We first consider the problem $\sum_{i=1}^{k} |r|_{[i]}$, which can be reformulated as:

$$\begin{array}{ll} \text{maximize} & |r|^T y, \\ \text{subject to} & 0 \succeq y \succeq 1, \mathbf{1}^T y = k \end{array}$$

which is equal to

$$\begin{array}{ll} \text{minimize} & -|r|^T y, \\ \text{subject to} & 0 \succeq y \succeq 1, \mathbf{1}^T y = k \end{array}$$

The *Lagrangian* is:

$$\begin{aligned} L(y, \lambda, u, t) &= -|r|^T y - \lambda^T y + u^T(y - \mathbf{1}) + t(\mathbf{1}^T y - k) \\ &= -\mathbf{1}^T u - kt + (-|r| - \lambda + u + t\mathbf{1})^T y \end{aligned}$$

Minimizing over y yields the dual function:

$$g(\lambda, u, t) = \begin{cases} -\mathbf{1}^T u - kt, & -|r| - \lambda + u + t\mathbf{1} = 0 \\ -\infty, & \text{otherwise} \end{cases}$$

The dual problem is to maximize g subject to $\lambda \succeq 0$ and $u \succeq 0$. Thus, we can express the dual problem as:

$$\begin{array}{ll} \text{maximize} & -\mathbf{1}^T u - kt, \\ \text{subject to} & -\lambda + u + t\mathbf{1} = |r| \\ & \lambda \succeq 0, u \succeq 0 \end{array}$$

which is equal to

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T u + kt, \\ \text{subject to} & u + t\mathbf{1} \succeq r \succeq -u - t\mathbf{1}, u \succeq 0 \end{array}$$

Then we combine with the original constraint, we can reformulate the problem as:

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T u + kt, \\ \text{subject to} & u + t\mathbf{1} \succeq Ax - b \succeq -u - t\mathbf{1}, u \succeq 0 \end{array}$$

with $u \in \mathbf{R}-$, $t \in \mathbf{R}$, and $x \in \mathbf{R}^n$.

A6.5

(a) We know that

$$\phi^{-1}(y_i) = a_i^T x + v_i, \quad i = 1, \ldots, m.$$

By definition of inverse function, we know that

$$\phi^{-1}(\phi(u)) = u \iff (\phi^{-1})'(\phi(u)) = \frac{1}{\phi'(u)}$$

and

$$\alpha \le \phi^{'}(u) \le \beta \implies \frac{1}{\beta} \le (\phi^{-1})'(y) \le \frac{1}{\alpha}$$

Therefore, $z_i = \phi^{-1}(y_i)$ and y_i have to satisfy the inequalities:

$$\frac{y_{i+1} - y_i}{\beta} \le z_{i+1} - z_i \le \frac{y_{i+1} - y_i}{\alpha}, \quad i = 1, \dots, m-1,$$

if we assume the points are sorted with y_i in increasing order. As the suggestion in the problem statement, we can use $z_1, \dots, z_m$ as parameters instead of ϕ. The log-likelihood function would be:

$$l(z, x) = \frac{-1}{2\sigma^2} \sum_{i=1}^{m} (z_i - a_i^T x)^2 - m \log(\sigma\sqrt{2\pi})$$

Therefore, in order to find a maximum likelihood estimate of x and z, we can solve the problem as:

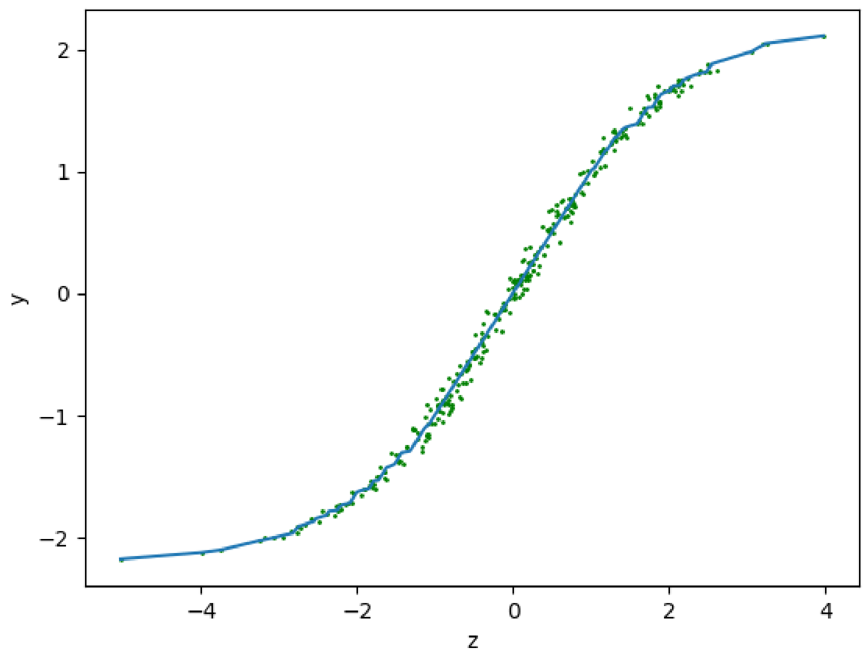
$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} (z_i - a_i^T x)^2, \\ \text{subject to} & \frac{y_{i+1}-y_i}{\beta} \le z_{i+1} - z_i \le \frac{y_{i+1}-y_i}{\alpha}, \quad i = 1, \dots, m-1 \end{array}$$

(b) The following Python code solve the problem:

```
import cvxpy as cvx
import matplotlib.pyplot as plt
import numpy as np

from nonlin_meas_data import *

x = cvx.Variable(n)
z = cvx.Variable(m)

B = np.zeros((m - 1, m))
for i in range(m - 1):
  B[i, i] = -1
  B[i, i + 1] = 1

nom_cost = cvx.Problem(cvx.Minimize(cvx.norm(z - A * x)),\
        [(1 / beta) * B @ y <= B @ z,\
        B @ z <= (1 / alpha) * B @ y]).solve(solver=cvx.CVXOPT)

print(x.value)

plt.figure()
```

```
plt.plot(z.value, y)
plt.scatter((A @ x).value, y, c = 'green', s = 1)
plt.xlabel('z')
plt.ylabel('y')
plt.savefig('A6.5.png')
plt.show()
```

The estimated x is x $= (0.48194444, -0.46569477, 0.93641226, 0.92966396)$. The blue curve in the figure shows the estimate function ϕ, and the green dots show the data points $a_i^T x, y_i$.



A.12.1d We can express the problem (c) for a given length N as the feasibility linear programming

$$0.89 \leq H(\omega_k) \leq 1.12 \quad \text{for} \quad 0 \leq \omega_k \leq \frac{\pi}{3},$$

$$-\alpha \leq H(\omega_k) \leq \alpha \quad \text{for} \quad \omega_c \leq \omega_k \leq \pi$$

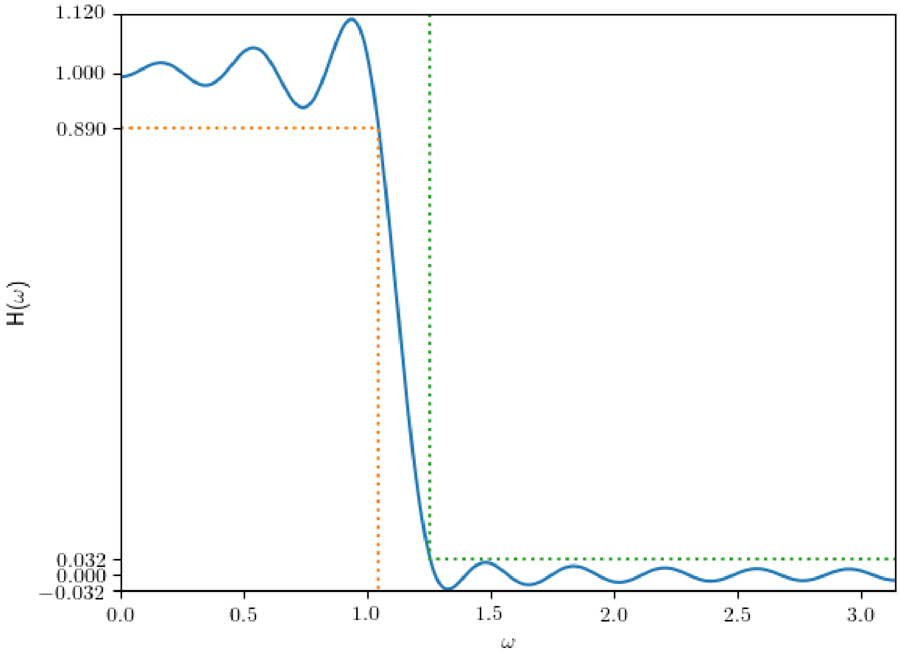
with variable a. The following Python code solves the quasi-convex optimization problem. We see that the shortest filter length is $N = 16$.

```
import numpy as np
```

```
import cvxpy as cvx
import matplotlib.pyplot as plt
from matplotlib import rc
rc('text', usetex = True)

K = 500
wp = np.pi / 3
wc = .4 * np.pi
alpha = 0.0316
w = np.linspace(0, np.pi, K).reshape((-1, 1))
wi = np.max(np.where(w <= wp)[0])
wo = np.min(np.where(w >= wc)[0])

H_final = None

for N in range(1, 50):
k = np.array(list(range(0, N + 1, 1))).reshape((-1, 1)).T
C = np.cos(w @ k)

a = cvx.Variable(N + 1)
problem = cvx.Problem(cvx.Minimize(0),\
[C[:wi, :] @ a <= 1.12,\
C[:wi, :] @ a >= 0.89,\
np.cos(wp * np.linspace(0, N, N + 1)) * a >= 0.89,\
C[wo:, :] @ a <= alpha,\
C[wo:, :] @ a >= -alpha,\
np.cos(wc * np.linspace(0, N, N + 1)) * a <= alpha])
problem.solve(solver=cvx.CVXOPT)
if problem.status == 'optimal':
print("The shortest filter length N = {}".format(N))
H_final = a.value.reshape((1, -1)) @ np.cos(w @ k).T
break

plt.figure()
plt.plot(w.T[0], H_final[0])
plt.plot([0, wp, wp], [0.89, 0.89, -alpha], ':')
plt.plot([wc, wc, np.pi], [1.12, alpha, alpha], ':')
plt.axis([0, np.pi, -alpha, 1.12])
plt.xlabel(r'$\omega$')
plt.ylabel(r'H($\omega$)')
plt.yticks([-alpha, 0, alpha, 0.89, 1, 1.12])
plt.savefig('A.12.d.png')
```

```
plt.show()
```

And the figure of the filter is:



A.15.1

(a) $A^T x$ gives a vector of the $x-$displacements of the springs, and $A^T y$ gives a vector of the $y-$displacements of the spring. Thus, we have the energy function as:

$$E(x, y, k) = \frac{1}{2} x^t A\mathbf{diag}(k) A^T x + \frac{1}{2} y^T A\mathbf{diag}(k) A^T y + c^T y$$

where $c_i = gm_i$. This is an affine function of k, so the minimum function over x and y that satisfy the fixed constraints. It follows that E_{min} is a concave function of k. Then we divide A into A_1 and A_2, where $A_1 \in \mathbf{R}^{p \times N}$ is made up of the frist p rows of A, and $A_2 \in \mathbf{R}^{(n-p) \times N}$ is made up of the last $n - p$ rows of A. Let $\bar{x}, \bar{y}, \bar{c} \in \mathbf{R}^{n-p}$ denote the last $n - p$ elements of x, y and c. Then the minimum energy can be expressed as:

$$E_{min}(k) = \min_{\bar{x}, \bar{y}} \left(\frac{1}{2} z_x^T \mathbf{diag}(k) z_x + \frac{1}{2} x_y^T \mathbf{diag}(k) z_y + \bar{c}^T \bar{y} + C \right)$$

where $C = \sum_{i=1}^{p} gm_i y_i^{fixed}$, $z_x = A_2^T \bar{x} + b_x$, $z_y = A_2^T \bar{y} + b_y$, $b_x = A_1^T x^{fixed}$, and $b_y = A_1^T y^{fixed}$. Thus to evaluate E_{min} we need to evaluate the minimum of an unconstrained quadratic in $\bar{x}$ and $\bar{y}$. This gives us

$$E_{min}(k) = \frac{1}{2}(b_x^T D b_x - v_x^T Q^{-1} v_x) + \frac{1}{2}(b_y^T D b_y - v_y^T Q^{-1} v_y) + C,$$

where $D = \mathbf{diag}(k)$, $Q = A_2 D A_2^T$, $v_x = A_2 D b_x$, and $v_y = A_2 D b_y + \bar{c}$.

We the reformulate the problem as:

$$\begin{array}{ll} \text{maximize} & b_x^T D b_x - v_x^T Q^{-1} v_x + b_y^T D b_y - v_y^T Q^{-1} v_y, \\ \text{subject to} & \mathbf{1}^T k = k^{tot}, k \succeq 0 \end{array}$$

This is a convex optimization problem. The constraints are convex in k. The objective function has two terms which are affine in k and two terms which are the negatives of matrix fractionals of affine terms in k. Thus, the objective is concave in k.

(b) The following code solves the problem:

```
tens_struct_data;

c = g * m
A1 = A(1:p,:)
A2 = A(p+1:n,:)
cbar = c(p+1:n)
cvx_begin
variable k(N)
D = diag(k);
bx = A1'*x_fixed;
by = A1'*y_fixed;
vx = A2*D*bx;
vy = A2*D*by + cbar;
maximize(bx'*D*bz - matrix_frac(vx, A2*D*A2') + ...
by'*D'by - matrix(vy,A2*D*A2'))
subject to
k >= 0; sum(k) == k_tot;
cvx_end

Eunif = 0.5 * x_unif' * A * diag(k_unif) * A' * x_unif;
Eunif = Eunif + 0.5 * y_unif' * A * diag(k_unif) * A' * y_unif';
Eunif = Eunif + c' * y_unif
Emin = 0.5 * cvx_optval + c(1:p)' * y_fixed

xmin = -(A2 * D * A2') \ (A2 * D * A1' * x_fixed);
ymin = -(A2 * D * A2') \ (A2 * D * A1' * y_fixed + cbar);
```

```
xopt = zeros(n, 1);
xopt(1:p) = x_fixed;
xopt(p + 1:n) = xmin;

yopt = zeros(n, 1);
yopt(1:p) = y_fixed;
yopt(p + 1: n) = ymin;
```

The optimal energy is $E_{min}(k^*) = 57.84$. The minimum energy is 18.37.

A.17.6

(a) We have $\phi_i'(u) = sin^{-1}(\frac{u}{k_j})$, which is a strictly increasing function of u, so ϕ_j is strictly convex. The optimality conditions for the problem are:

$$Ap^* = s, \quad \nabla\phi(p^*) - A^T\nu^* = 0.$$

The second equation can be expressed as

$$\phi'(j(p_j^*) = a_j^T\nu^*, \quad j = 1, \ldots, m,$$

where a_j is the jth column of A. Thus, we have

$$\sin^{-1}(\frac{p_j^*}{\kappa_j}) = a_j^T\nu^*, \quad j = 1, \ldots, m$$

Thus,

$$p_j^* = \kappa_j \sin(a_j^T\nu^*), \quad j = 1, \ldots, m$$