Sistemas Discretos - 2017.1

Departamento de Eletrônica & Sistemas - UFPE
Sistemas Discreto - Projeto 2

I. Introdução

O Projeto vale 20% da nota da unidade. Os melhores projetos para cada atividade específica terão direito de fazer uma apresentação. As melhores apresentações ganham até 1 (um), 0,5 (meio) ou 0,25 (quarto) adicional na média da unidade, respectivamente. Haverá apenas uma apresentação para cada atividade específica. As equipes devem conter exatamente 3 alunos. Os projetos devem ser entregues em versão impressa (ou escrita) até o dia 07/12/2017 no inicio da aula. Será avaliado no projeto: a) Entrega no prazo, b) Quantidade correta de membros na equipe, c) Resultados obtidos, d) Originalidade e e)Organização, encadeamento lógico e escrita. A nota será dada por $N_p = N_a N_b (N_c + N_d + N_e + N_f)/400$.

II. Atividades básicas do Projeto

As atividades básicas podem ser realizadas por todas as equipes como preparação para as atividades específicas. Não é necessário a inclusão das atividades básicas no projeto.

A. Processamento de imagem

1) O programa Scilab oferece vários módulos adicionais com aplicações em diversas áreas do conhecimento científico. Alguns desses módulos podem ser encontrados em (www.scilab.org/scilab/modules) e (atoms.scilab.org). Nesse projeto, será utilizado o *Scilab Image and Video Processing toolbox* (SIVP). Instale o SIVP a partir do comando {atomsInstall('SIVP')} na linha de comando do Scilab (necessita de conexão com a internet);
2) Utilize a função *imread()* para abrir o arquivo "barco.png" (disponível no grupo de Sistemas Discretos). Utilize as funções *im2double()* e *rgb2gray()* para converter a imagens para ponto flutuante com escala cinza (valores de 0 até 1). Utilize a função *imshow()* para visualizar as imagens;
3) Utilize a função *mat2gray* para converter uma matriz usual do Scilab para uma imagem em escala cinza. Construa e mostre as seguintes imagens, $x_i[m,n]$, $m,n = 0,1,\ldots,255$. a) $x_1[m,n] = n$; b) $x_2[m,n] = m$; c) $x_3[m,n] = 255e^{-[(m-128)^2+(n-128)^2]/8192}$. Deve-se utilizar a função *mat2gray()* antes de *imshow()* para converter as matrizes em imagens.

III. Atividades Específicas do Projeto

Cada equipe deve realizar apenas uma das atividades específicas descritas a seguir. O relatório do projeto deve ser feito baseado na atividade específica.

A. Curvas Elípticas sobre Corpos Finitos

Criptografia baseada em curvas elípticas foi proposta inicialmente em 1985 por Vic Miller e Neal Koblitz. Uma curvas elíptica (CE) é o conjunto de todas as soluções $(x,y) \in \mathbb{Z}_p^2$, $p > 2$ um número primo, de $y^2 = x^3 + ax + b \pmod p$, $a,b \in \mathbb{Z}_p$ tais que $4a^3 + 27b^2 \neq 0 \pmod p$, mais um ponto chamado de Infinito, denotado por $\vec{\Im}$.

Definição 1: A soma de dois pontos, $\vec{p} = (x_p, y_p)$ e $\vec{q} = (x_q, y_q)$, de uma CE, realizada através da operação binária denotada por $\oplus$, é definida como $\vec{r} = (x_r, y_r)$, em que

- Se $x_p = x_q$ e $y_p = -y_q$, então $\vec{r} = \vec{\Im}$;
- Senão, $x_r = t^2 - x_p - x_q$, $y_r = t(x_p - x_r) - y_p$, em que

$$t = \begin{cases} (y_q - y_p)(x_q - x_p)^{-1}, & \text{se } \vec{p} \neq \vec{q}, \\ (3x_p^2 + a)(2y_p)^{-1}, & \text{se } \vec{p} = \vec{q}. \end{cases}$$

- para todo $\vec{p}$ pertencendo a curva elíptica,

$$\vec{p} \oplus \vec{\Im} = \vec{\Im} \oplus \vec{p} = \vec{p}.$$

Nesse contexto:

1) Prove que $\langle \text{CE}, \oplus \rangle$ é um grupo.
2) Implemente a operação $\oplus$ entre pontos de uma CE considerando p com cerca de 512 bits.
3) Implemente um algoritmo rápido para calcular

$$\vec{p}^N = \underbrace{\vec{p} \oplus \vec{p} \ldots \oplus \vec{p}}_{N \text{ vezes}}.$$

   Qual o número máximo de adições necessárias em função da quantidade de bits de N?
4) Faça um breve estudo a sobre a ordem do grupo CE e a segurança de um sistema com segurança baseada no logaritmo discreto sobre curvas elípticas.
5) Implemente o ElGamal sobre curvas elípticas ou o sistema Diffie-Hellman.

B. Análise de desempenho para códigos de canal

Um canal de comunicação BSC é um modelo digital para uma comunicação com uma modulação binária assumindo um ruído aditivo com distribuição de probabilidade gaussiana. Nesse modelo, cada bit transmitido tem a probabilidade p de ser modificado (1 muda para 0 ou 0 muda para 1), em que

$$p = Q\left(\sqrt{\frac{2RE_b}{N_0}}\right),$$

em que E_b/N_0 é a relação sinal ruído (SNR) do canal, $R = k/n$ é a taxa de informação código utilizada na transmissão ($R = 1$ se nenhum código é utilizado) e a função $Q(x)$ é dada por

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt.$$

Nesse contexto, analise o desempenho de alguns códigos de bloco.

1) Implemente os codificadores para os códigos binários de Hamming $C_1(7,4,3)$, $C_2(15,11,3)$ e para o código de repetição $C_3(5,1,5)$ e os seus respectivos decodificadores.
2) Implemente uma função noise(n,p) que retorna um vetor de ruído $\vec{e}$ aleatório com comprimento n e com probabilidade de cada componente ser 1 igual de p.
3) A relação sinal ruído do canal SNR pode ser dada em DB através de

$$\text{SNR}_{\text{DB}} = 10\log(\text{SNR}).$$

Construa um gráfico com as probabilidade de erro de bit p em função de SNR_{DB}, $3 \leq \text{SNR}_{\text{DB}} \leq 10$, para um código de taxa $R = 1$, $R = 4/7$, $R = 11/15$ e $R = 1/5$, no mesmo gráfico. O eixo da probabilidade deve ser representado em logaritmo.
4) Considerando a probabilidade de erro para cada código, faça simulações de várias transmissões e construa um gráfico da probabilidade de erro, após a decodificação do respectivo código, para cerca de 20 valores diferentes de SNR_{DB} entre $3 \leq \text{SNR}_{\text{DB}} \leq 10$.
5) Sugestão 1: para um dado código, para cada valor diferente de SNR_{DB}, calcule p, escolha $\vec{m}$ (ex: $\vec{m} = \vec{0}_k$), encontre $\vec{c} = \vec{m}G$ ($\vec{c} = \vec{0}_n$), e $\vec{r} = \vec{e} + \vec{c}$ ($\vec{r} = \vec{e} = \text{noise}(n,p)$), e decodifique $\vec{r}$ para obter $\vec{m}_r$. O número de erros é a quantidade de uns em $\vec{m} - \vec{m}_r$ ($W_H(\vec{m}_r)$). Acumule a quantidade de erros E obtidos em M simulação até que esse número seja $E \geq N$ (ex: N = 100). A probabilidade de erros de bit do código é dada aproximadamente por $E/(kM)$.
6) Sugestão 2: guarde o resultado das simulações em arquivos diferentes para cada código para que os dados possam ser usados depois sem a necessidade de nova simulação.
7) Avalie o resultado das curvas construídas no gráfico. Qual dos códigos implementados apresenta o melhor desempenho?

C. Códigos de canal aplicado a imagem

Imagens coloridas com resolução $M \times N$ são representadas por 3 matrizes $A^{(k)} = [a_{ij}^{(k)}]$, $M \times N$, em que cada componente a_{ij} representa a intensidade do vermelho (R), do verde (G) ou do azul (B) da imagem. Assim, $k = 0, 1, 2$ ou $k = r, g, b$, em que cada componente $a_{ij}^{(k)}$ é representada por um byte (um conjunto de 8 bits). Nesse contexto:

1) Implemente uma função img2vecbin(img) que transforma uma imagem RGB com resolução $M \times N$ e com 24 bits de codificação de pixel (8 bits para cada cor) para um vetor binário de comprimento $24MN$. Construa a função inversa vecbin2img$(vecbin, M, N)$, que converte um vetor binário novamente em uma imagem RBG.
2) Implemente uma função noise(N,p) que gera um vetor binário de comprimento N e com probabilidade de cada bit ser 1 igual a p.
3) Simule o efeito do ruído aditivo em uma imagem colorida com probabilidades p de 0,05; 0,01; 0,001 e 0,0001. Sugestão: Transforme uma imagem em um vetor com $\vec{m} = \text{img2vecbin}(img)$, construa o ruído com $\vec{e} = \text{noise}(24MN, p)$, aplique um ou-exclusivo para simular o efeito do ruído $\vec{r} = \vec{m} + \vec{e}$ e utilize vecbin2img para ver a imagem.
4) Implemente um código de Hamming (15, 11, 3) para calcular as redundâncias de cada 11 bits um vetor e um decodificador para corrigir possíveis erros utilizando os bits do vetor e das redundâncias. Quantos bits de redundância são necessários?
5) Simule o efeito do ruído em uma imagem colorida e em sua redundância com as probabilidades p de 0,05; 0,01; 0,001 e 0,0001. A partir do resultado da simulação, aplique o decodificador do código de Hamming para obter uma nova imagem utilizando a imagem e redundância corrompidas.
6) Compare as imagens decodificadas com as imagens corrompidas para cada valor de probabilidade. Qual o aumento do consumo de energia (percentual) na transmissão da mensagem e da redundância quando comparado com a transmissão apenas da imagem? Encontre uma formula em função de $R = k/n$.
7) Interprete os resultados obtidos.

IV. Relatório do Projeto

O relatório deve conter os tópicos das Atividades Específicas do Projeto de sua equipe e deve ser organizado nas seguintes seções:

1) Resumo (Opcional, não é seção): Deve conter um breve resumo sobre o trabalho com os principais resultados obtidos;
2) Introdução/Motivação: Deve conter uma breve introdução/motivação sobre o projeto, com objetivos e roteiros;
3) Fundamentação Teórica: Contém um resumo da teoria utilizada no projeto;
4) Metodologia: Contém a solução das atividades propostas no projeto e os resultados das simulações;
5) Análise: Contém a análise dos resultados obtidos em comparação com os resultados teóricos. Quaisquer dificuldades ou problemas encontrados devem ser relatados nessa seção;
6) Conclusão ou Conclusões: Contém a/as conclusão/conclusões do relatório, um resumo de tudo que foi feito e aprendido, comentários sobre os modelos utilizados e possíveis aplicações.
7) Referências (Opcional): Essa seção deve conter as referências citadas no decorrer do projeto.
8) Apêndice: Coloque as Atividades Básica do Projeto nesta parte do relatório.
9) Anexo (Opcional): Cópia de códigos fontes de algoritmos.